INTERACTIVE MACHINE LEARNING: FROM THEORY TO SCALE

by

Yinglun Zhu

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2023

Date of final oral examination: June 06, 2023

The dissertation is approved by the following members of the Final Oral Committee:
Robert D. Nowak, Professor, Electrical & Computer Engineering, UW–Madison
Kevin Jamieson, Assistant Professor, Computer Science & Engineering, University of Washington
Kangwook Lee, Assistant Professor, Electrical & Computer Engineering, UW–Madison
Rebecca Willett, Professor, Statistics and Computer Science, University of Chicago
Stephen J. Wright, Professor, Computer Sciences, UW–Madison
Xiaojin (Jerry) Zhu, Professor, Computer Sciences, UW–Madison

To mom and dad.

Essentially, all models are wrong, but some are useful.

— George E. P. Box, University of Wisconsin–Madison

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my Ph.D. advisor Robert D. Nowak, for his continuous mentoring, support, and encouragement. Rob gives me a great freedom to explore topics I am interested in, but, at the same time, he is always passionate about discussing research problems with me and helping me get out of trouble. Rob has been everything I could ask for as an advisor. Rob is also a role model to me, as a researcher and mentor, who will keep motivating me in my future academic career.

I had a wonderful summer intern at Microsoft Research NYC in 2021, where I was fortunate to be mentored by Dylan J. Foster, John Langford, and Paul Mineiro. I am extremely grateful to their mentoring: They not only teach me how to approach difficult research problems, but also spent their time helping me sharpen my presentation and communication skills.

I am thankful to my committee members: Kevin Jamieson, Kangwook Lee, Rebecca Willett, Stephen J. Wright, and Xiaojin (Jerry) Zhu. They have been great information sources for me during my Ph.D. journey, and have consistently provided me with invaluable advice and insightful feedback.

I was fortunate to have collaborated with many outstanding researchers over the past six years: Gregory Canal, Yifang Chen, Simon S. Du, Dylan J. Foster, Quanquan Gu, Kevin Jamieson, Ruoxi Jiang, Sumeet Katariya, Julian Katz-Samuels, John Langford, Paul Mineiro, Stephen Mussmann, Robert D. Nowak, Mark Rucker, Rebecca Willett, Jifan Zhang, and Dongruo Zhou. I am grateful to all of them for their patience, encouragement, and friendship. The contents of this dissertation have benefited especially from collaborations with Dylan J. Foster, Julian Katz-Samuels, John Langford, Paul Mineiro, and Robert D. Nowak.

Studying at UW–Madison has been a great experience. I would like to thank my labmates for their support and encouragements: Gregory Canal, Danica Fliss, Mina Karzand, Sumeet Katariya, Julian Katz-Samuels, Jeongyeol Kwon, Blake Mason, Haley Massa, Subhojyoti Mukherjee, Julia Nakhleh, Rahul Parhi, Joseph Shenouda, Scott Sievert, Gokcan Tatli, Ardhendu Tripathy, Liu Yang, and Jifan Zhang. I would

also like to extend my sincere thanks to all my friends who have supported me over the past six years. This acknowledgement is way too short to list all the names, but you know who you are.

Finally, I would like to thank my parents Meirong and Xiangzhong: None of this would have been possible without their unconditional love and support.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

While machine learning has made unprecedented successes in many real-world scenarios, most learning approaches require a huge amount of training data. Such a requirement imposes real challenges to the practitioners, e.g., data annotation can be expensive and time-consuming. To overcome these challenges, this dissertation studies interactive machine learning, where learning is conducted in a closed-loop manner: The learner uses previously collected information to guide future decisions (e.g., which data points to label next), which in turn help make the following predictions.

This dissertation focuses on developing novel algorithmic principles and uncovering fundamental limits when scaling interactive machine learning into real-world settings at large scales. More specifically, we study interactive machine learning with (i) noisy data and rich model classes, (ii) large action spaces, and (iii) model selection requirements; this dissertation is thus grouped into three corresponding parts. To bring the promise of interactive learning into the real world, we develop novel human-in-the-loop learning algorithms and systems that achieve both statistical efficiency and computational efficiency.

In the first part, we study active machine learning with noisy data and rich model classes. While huge successes of active learning have been observed, due to technical difficulties, most guarantees are developed (i) under low-noise assumptions, and (ii) for simple model classes. We develop efficient algorithms that bypass these two fundamental barriers and thus make an essential step toward real-world applications of active learning. More specifically, by leveraging the power of abstention, we develop the first efficient, general-purpose active learning algorithm that achieves exponential label savings without any low-noise assumptions. We also develop the first deep active learning (i.e., active learning with neural networks) algorithms that achieve exponential label savings when equipped with an abstention option.

In the second part, we study decision making with large action spaces. While researchers have explored decision making when the number of alternatives (e.g., actions) is small, guarantees for decision making in large, continuous action spaces

remained elusive, leading to a significant gap between theory and practice. In this part, we bridge this gap by developing the first efficient, general-purpose contextual bandits algorithms for large action spaces, in both structured and unstructured cases. Our algorithms make use of standard computational oracles, and achieve nearly optimal guarantees, and have runtime and memory independent of the size of the action space. Our algorithms are also highly practical: They achieve the state-of-the-art performance on an Amazon dataset with nearly 3 million categories.

In the third part, we study model selection in decision making. Model selection is the fundamental task in supervised learning, but it faces unique challenges when deployed in decision making: Decisions are made online and only partial feedback is observed. Focusing on the regret minimization setting, we establish fundamental lower bounds showing that model selection in decision making is strictly harder than model selection in standard supervised learning: Compared to an additional logarithmic cost suffered in supervised learning, one has to pay an additional polynomial cost in decision making. Nevertheless, we develop Pareto optimal algorithms that achieve matching guarantees (up to logarithmic factors). Focusing on the best action identification setting, we develop novel algorithms and show that model selection in best action identification can be achieved without too much additional cost.

1 OVERVIEW

1.1 Introduction

In the past decade, we have witnessed huge successes of machine learning in many fields, including image recognition (Krizhevsky et al., 2012; LeCun et al., 2015), natural language processing (Bahdanau et al., 2014; Brown et al., 2020), game playing (Silver et al., 2016; Berner et al., 2019), and many others. The formula of machine learning, however, it not complicated: Focusing on classical statistical learning, the learner's goal is to learn patterns from collected datasets in order to make future predictions. We now discuss a concrete success for this formula: In 2009, Dr. Fei-Fei Li and her team curated a large-scale image classification dataset ImageNET with around 15,000,000 labeled images across 22,000 categories (Russakovsky et al., 2015). With the availability of this huge dataset, after many years of research (e.g., developing better models and optimizers), machine learned classifiers eventually achieve super-human image classification accuracy(Russakovsky et al., 2015; Krizhevsky et al., 2012; He et al., 2016).

This achievement is remarkable, however, one should notice that the labeling process of ImageNET is not easy at all: It took 48,940 human annotators from Amazon Mechanical Turk across 167 countries more than 2 years. This data labeling process is clearly expensive and time-consuming, and it's not hard to see that one cannot afford to scale such efforts to every real-world applications, especially for those in medical and robotics domains. As a result, data labeling is increasingly becoming the bottleneck for real-world machine learning deployments.

To address the aforementioned bottleneck, this dissertation studies interactive machine learning, where the learner uses previously collected information to guide future decisions (e.g., which data points to label next), which in turn help make the following predictions. By iteratively deploying this closed-loop learning procedure, our hope is to conduct learning in a much more efficient way (e.g., learn with much fewer labels).

Two interactive machine learning paradigms are considered in this dissertation:

prediction and *decision making*. Focusing on prediction, interactive machine learning specializes to active learning, where the learner actively select which data points to label so that they can learn an accurate classifier/regression function with less labeled data points. Focusing on decision making, the learner interactively selects the actions to take in order to minimize regret or efficiently identify the best action.

While there have been many prior work on interactive machine learning, this dissertation focuses on developing novel algorithmic principles and uncovering fundamental limits when scaling interactive machine learning into real-world settings at large scales. Three main themes of the dissertation include:

- **Learning with noisy data and rich model classes.** Most favorable guarantees for interactive learning have been developed (i) under low-noise assumptions, and (ii) for simple model classes. To scale interactive learning into real-world settings, one central theme of this dissertation is to develop general algorithmic principles that work with noisy data and rich model classes (such as deep neural networks).

- **Learning with large action spaces.** Learning with small action spaces is relatively easy since one can uniformly explore all the actions and suffer an additional cost the scales with the number of actions. Such uniform strategy, however, will fail in the large action settings with potentially millions of actions. The second central theme of this dissertation is to develop novel learning algorithms that work with large action spaces.

- **Learning with model selection.** Model selection has been virtually used in every machine learning pipelines to select the best model class; it is also theoretically well-understood in classical supervised learning through the structure risk minimization principle. However, model selection was less-studied in interactive machine learning. The third main theme of this dissertation is to understand the fundamental limits of model selection in interactive learning.

To bring the promise of interactive machine learning into practice, we focus on developing novel human-in-the-loop learning algorithms and systems that

achieve both *statistical efficiency* and *computational efficiency*. Statistically, we aim at establishing guarantees that match the fundamental lower bounds. Computationally, we focus on developing algorithms that can be implemented with efficient computational primitives.

In the next few sections, we introduce basic learning paradigms that feature throughout this dissertation. We provide the organization of the dissertation in Section 1.4, and the bibliographic details in Section 1.5. We introduce the basic notation in Section 1.6.

1.2 Passive and Active Learning

Machine learning focuses on the use of data and algorithms to imitate the way that human learn. Focusing on prediction tasks, the learner tries to learn a *classifier* $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the *instance space*, and $\mathcal{Y}$ is the *label space*. We primarily consider the classical binary classification tasks where the label space is $\mathcal{Y} := \{+1, -1\}$. The joint distribution over $\mathcal{X} \times \mathcal{Y}$ is denoted as $\mathcal{D}_{\mathcal{X}\mathcal{Y}}$. We use $\mathcal{D}_{\mathcal{X}}$ to denote the marginal distribution over the input space $\mathcal{X}$, and use $\mathcal{D}_{\mathcal{Y}|x}$ to denote the conditional distribution of $\mathcal{Y}$ with respect to any $x \in \mathcal{X}$. For any classifier $h : \mathcal{X} \to \mathcal{Y}$, its error is defined as $\mathrm{err}(h) := \mathbb{P}_{(x,y)\sim\mathcal{D}_{\mathcal{X}\mathcal{Y}}}(h(x) \neq y)$. Consider a *hypothesis class* $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$, we use $h^\star \in \mathcal{H}$ to denote the classifier within the hypothesis class that achieves the smallest error, i.e., $h^\star := \arg\min_{h\in\mathcal{H}} \mathrm{err}(h)$. For any classifier $h : \mathcal{X} \to \mathcal{Y}$, we use $\mathrm{excess}(h) := \mathrm{err}(h) - \mathrm{err}(h^\star)$ to denote the *excess error* of h. The learner's goal is to learn a classifier with small excess error: Learning is usually considered in the *Probably Approximately Correct* (PAC) setting (Vapnik and Chervonenkis, 1971; Vapnik, 1995; Valiant, 1984; Haussler, 1992): Given parameters $\varepsilon > 0, \delta \in (0, 1)$, the learner's goal is to, with probability at least $1 - \delta$, identify a classifier $\widehat{h}$ such that

$$\mathrm{err}(\widehat{h}) \leqslant \mathrm{err}(h^\star) + \varepsilon. \tag{1.1}$$

Passive learning. We use the term *passive learning* to refer to the classical *supervised/statistical learning* approach, in order to distinguish it from the *active learning*

approach that will discussed shortly after. In passive learning, the learner collects a dataset $\{(x_i, y_i)\}_{i=1}^n$ that is independently and identically distributed (i.i.d.) generated from the joint distribution $\mathcal{D}_{\mathcal{X}\mathcal{Y}}$, and then learns a classifier $\widehat{h} : \mathcal{X} \to \mathcal{Y}$. The number of data points n needed to satisfy the requirement in Eq. (1.1) is referred to as the *sample complexity*. Passive learning has been extensively studied in the literature and now well-understood: A hypothesis class $\mathcal{H}$ is PAC learnable if and only if $\mathcal{H}$ has finite VC dimension $\mathrm{VCdim}(\mathcal{H})$—a complexity measure that characterizes the complexity of the hypothesis class (Vapnik and Chervonenkis, 1971; Shalev-Shwartz and Ben-David, 2014). Assuming finite VC dimension, the sample complexity of passive learning scales as $\widetilde{\Theta}(\mathrm{VCdim}(\mathcal{H}) \cdot \mathrm{poly}(\frac{1}{\varepsilon}))$, i.e., it is polynomial in terms of $\frac{1}{\varepsilon}$.

Active learning. In active learning, different from passive learning, the dataset is not i.i.d. generated but interactively collected. More specifically, the learner has access to a labeling oracle, where the learner can call the labeling oracle with any *unlabeled* data point x as input and get back its label $y \sim \mathcal{D}_{\mathcal{Y}|x}$. Instead of labeling all unlabeled data points, the active learner interactively selects data points to be labeled next, based on previously collected information. The hope here is that, compared to the passive learner, the active learner can learn a classifier satisfying Eq. (1.1) with much fewer labeled data points. We define *label complexity* as the number of calls to the labeling oracle, and measure the performance of the active learner in terms of label complexity.

A canonical example to show the active learning gain over passive learning is learning a one dimensional threshold function in the noiseless case. Passive learning requires $\Omega(\frac{1}{\varepsilon})$ labels in this case, but active learning—instantiated through binary search—can learn the target function with only $O(\log \frac{1}{\varepsilon})$ labels, which exhibits a remarkable *exponential speedup* over passive learning. Beyond learning a simple threshold function, in the past few years, researchers have established positive active learning results when learning from other hypothesis classes (Balcan et al., 2007; Hanneke, 2007; Dasgupta et al., 2009; Hsu, 2010; Dekel et al., 2012; Hanneke, 2014; Zhang and Chaudhuri, 2014; Krishnamurthy et al., 2019; Katz-Samuels et al., 2021).

Exponential speedups over passive learning, however, are mainly observed for certain (simple) hypothesis classes (e.g., linear classifiers) under certain low-noise assumptions (e.g., Massart noise, Massart and Nédélec (2006)).

1.3 Sequential Decision Making

Besides prediction tasks, another interesting problem in machine learning is sequential decision making, where the learner make decisions on the fly and sequentially observe feedback. More specifically, the learner is given an action set $\mathcal{A}$, and the decision making process proceeds in rounds. At each round, the learner receives a context x_t, takes an action $a_t \in \mathcal{A}$, and then observes a reward r_t. In this dissertation, we primarily focus on the bandit problem where the observed reward $r_t = r_t(a_t)$ only corresponds to the taken action a_t but no other actions—the so-called learning bandit feedback setting. For bandit problems, the actions are sometimes referred to as arms, corresponding to the arms of the slot machines (Bubeck and Cesa-Bianchi, 2012). When the context x_t observed in each round remains constant, the contextual bandit problem is reduced to the non-contextual bandit problem. Depending on the goal of the learner, we divide the bandit problems into two categories: *regret minimization* and *best action identification*, which we described below separately.

Regret minimization. In regret minimization, the decision making process proceeds for T rounds. The learner is given a policy class Π, where each policy $\pi \in \Pi$ is a mapping from the context space $\mathcal{X}$ to the action space $\mathcal{A}$, i.e., $\pi : \mathcal{X} \to \mathcal{A}$. Let $\pi^\star := \arg\max_{\pi \in \Pi} \sum_{t=1}^{T} r_t(\pi(x_t))$ denote the optimal policy in the hindsight. The goal of the learner is to minimize the cumulative regret (or its expected version), which is defined as

$$\mathbf{Reg}(T) := \sum_{t=1}^{T} r_t(\pi^\star(x_t)) - r_t(a_t). \tag{1.2}$$

The notion regret in Eq. (1.2) captures the difference between the learner's performance and the performance of the optimal policy $\pi^\star$; in order words, we measure the *regret* of the learner for not playing optimally (Bubeck and Cesa-Bianchi, 2012). The regret minimization framework has been extensively deployed in practice for online personalization and recommendation tasks (Li et al., 2010; Agarwal et al., 2016; Tewari and Murphy, 2017; Cai et al., 2021). Any regret that scales sublinearly in T, i.e., $\mathbf{Reg}(T) = o(T)$, implies that, in the long run, the learner's behavior converges to the behavior of the optimal policy $\pi^\star$. A golden standard for regret minimization is to achieve regret guarantees scales as $\mathbf{Reg}(T) = \widetilde{\Theta}(\sqrt{T})$, which has been proved to be the optimal guarantee one can hope for (Agarwal et al., 2012, 2014; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021).

Best action identification. In best action identification, the learner's goal is to efficiently identify the action that (approximately) achieves the highest reward. Focusing on the non-contextual setting with stochastic rewards, the optimal action $a^\star \in \mathcal{A}$ is defined as

$$a^\star := \arg\max_{a \in \mathcal{A}} \mathbb{E}_r[r(a)]. \tag{1.3}$$

The best action identification framework has been extensively deployed on online crowdsourcing and biomedical domains (Zhou et al., 2014; Tanczos et al., 2017; Réda et al., 2020; Aziz et al., 2021). Two different settings are studied in best action identification: The *fixed confidence setting* and *the fixed budget* setting. In the fixed confidence setting, given a confidence parameter $\delta \in (0, 1)$, the learner tries to identify the best action $a^\star$ (or a near-optimal action) with probability at least $1 - \delta$ while minimizing the number of samples (Mannor and Tsitsiklis, 2004; Even-Dar et al., 2006). In the fixed budget setting, given a budget T on the number of samples, the learner outputs an action $\widehat{a}$ and minimize the probability that such action is not the optimal action $a^\star$, i.e., minimize the probability $\mathbb{P}(\widehat{a} \neq a^\star)$ (Hoffman et al., 2014; Katz-Samuels et al., 2020). In both settings, the learner's goal is to adapt the fundamental instance-dependent complexity—a complexity adapts to the given

problem instance—rather than the usual worst-case complexity.

1.4 Highlights and Organization

Chapter 1 gives an overview of this dissertation, where we introduce basic settings in interactive machine learning and highlights the main contributions. From here on, this dissertation is broken into three parts.

Part I: Active Learning with Noisy Data and Rich Model Classes. Active learning becomes increasingly important in modern applications since unlabeled data points are abundant, yet the labeling process is expensive and time-consuming. However, due to technical difficulties, previous active learning guarantees were mainly developed (i) in noiseless or low-noise settings and (ii) for simple models such as threshold functions and linear classifiers. In Part I, we develop efficient algorithms that bypass these two fundamental barriers and thus make an essential step towards real-world applications of active learning.

Researchers have long been analyzing active learning under low-noise assumptions (e.g., Massart and Tsybakov noises) due to a fundamental lower bound showing that active learning provides no gains over passive learning in high-noise regimes. To jump out of the box, in Chapter 2, we explore active learning with an additional abstention option, i.e., whenever the classifier abstains, it incurs a cost marginally smaller than random guessing (formalized as Chow's error). With access to a supervised convex loss regression oracle (e.g., least squares for linear models), we develop the first computationally efficient active learning algorithm that achieves exponential label savings without any low-noise assumptions. The developed result is not only theoretically exciting but also practically meaningful. For instance, in medical domains, it's ideal to defer high-risk decisions to experts if the classifier is uncertain about its own predictions. We also develop novel extensions of the main result, e.g., recovering minimax optimal results in the standard setting, and achieving *constant* label complexity for finite hypothesis classes.

To move one step further towards real-world applications of active learning, in Chapter 3, we study active learning with neural networks (also known as deep active learning). While many researchers have empirically explored deep active learning, its theoretical foundations remained elusive. By carefully trading-off approximation error and learning error, we develop the first deep active learning algorithm that achieves nearly minimax optimal label complexity guarantees. When additionally equipped with the abstention option, we develop the first deep active learning algorithm that achieves exponential savings in label complexity. Our results are derived by building a general bridge between approximation theory and active learning guarantees, which is of independent interest. Our results provide theoretical justifications for many existing deep active learning approaches that achieve impressive empirical performance.

Part II: Decision Making with Large Action Spaces. While researchers have explored decision making when the number of alternatives (e.g., actions) is small, guarantees for decision making in large, continuous action spaces remained elusive, leading to a significant gap between theory and practice. In Part II, we bridge this gap by tackling large-scale decision making problems in both *structured* and *unstructured* cases.

In Chapter 4, we study the structured case and develop the first efficient, general-purpose algorithm for contextual bandits with continuous, linearly structured action spaces. The developed algorithm makes use of standard computational oracles for (i) supervised learning, and (ii) *linear* optimization over the action space; it achieves nearly optimal guarantees, and has runtime and memory independent of the size of the action space. Our algorithm is also highly practical: it achieves state-of-the-art performance on an Amazon dataset with nearly 3 million categories.

The unstructured decision making problems are generally intractable since—with unstructured regression function classes—one can easily construct situations where the learner has to "identify a needle in the haystack". To overcome such pathological examples, in Chapter 5, we study the unstructured case with smoothed benchmarks, i.e., competing against a smoothed distribution rather than a Dirac

delta distribution concentrating on the needle-in-the-haystack point. Focusing on contextual bandits, we develop the first efficient, general-purpose algorithm that works with *any* unstructured regression functions (as long as they are measurable). Our algorithm achieves the optimal regret guarantees; when additional structural assumptions exist (e.g., under Lipschitz or Hölder continuity), it also recovers existing minimax results when compete against the standard non-smoothed benchmark.

Part III: Model Selection in Decision Making. Model selection is the fundamental statistical task of adapting to the right hypothesis class using data, and it has been used in virtually every machine learning pipeline. However, decision making presents special challenges to model selection since decisions are made online, and only partial feedback is observed. In Part III, we uncover the fundamental limits for model selection in decision making and develop efficient algorithms that achieve near-optimal performance.

We first study the regret minimization problem in decision making. In Chapter 6, we study model selection in the unstructured case, where there exist multiple best actions in the action set and one hope to scale the regret as the effective number of actions, rather than the total number of actions. In Chapter 7, we study model selection in the structured case, where there is a nested sequence of linear hypothesis classes and one hope to adapt to the complexity of the smallest hypothesis class containing the true function. In both cases, we establish fundamental lower bounds showing that model selection in decision making is strictly harder than model selection in standard supervised learning: Compared to an additional logarithmic cost suffered in supervised learning, one has to pay an additional *polynomial cost* in decision making. Nevertheless, we develop *Pareto optimal* algorithms that achieve guarantees matching the lower bounds, up to logarithmic factors. A different Pareto optimal model selection algorithm is also provided and analyzed in Chapter 5.

In Chapter 8, we introduce and study model selection in the best action identification setting, where we consider both the fixed confidence setting and the fixed budget setting. Given a nested sequence of hypothesis classes of increasing complexities, our goal is to automatically adapt to the instance-dependent complexity

measure of the smallest hypothesis class containing the true model, rather than suffering from the complexity measure related to the largest hypothesis class. We develop algorithms that solve a novel optimization problem based on experimental design that leverages the geometry of the action set to efficiently identify a near optimal hypothesis class. Different from the regret minimization problems, we show that model selection in best action identification can be achieved without too much additional cost.

1.5 Bibliographic Notes

Results in Part I are based on joint work with Robert D. Nowak:

- Yinglun Zhu and Robert D. Nowak. 2022. Efficient active learning with abstention. *Advances in Neural Information Processing Systems.*
- Yinglun Zhu and Robert D. Nowak. 2022. Active learning with neural networks: Insights from Nonparametric Statistics. *Advances in Neural Information Processing Systems.*

Results in Part II are based on joint work with Dylan J. Foster, John Langford, and Paul Mineiro:

- Yinglun Zhu and Dylan J. Foster, John Langford, and Paul Mineiro. 2022. Contextual bandits with large action spaces: Made practical. *International Conference on Machine Learning.*
- Yinglun Zhu and Paul Mineiro. 2022. Contextual bandits with smooth regret: Efficient learning in continuous action spaces. *International Conference on Machine Learning.*

Results in Part III are based on joint work with Julian Katz-Samuels and Robert D. Nowak:

- Yinglun Zhu and Robert D. Nowak. 2020. On regret with multiple best arms. *Advances in Neural Information Processing Systems.*
- Yinglun Zhu and Robert D. Nowak. 2022. Pareto optimal model selection in linear bandits *International Conference on Artificial Intelligence and Statistics.*
- Yinglun Zhu, Julian Katz-Samuels, and Robert D. Nowak. 2022. Near instance optimal model selection for pure exploration linear bandits *International Conference on Artificial Intelligence and Statistics.*

1.6 Notation

We define some general notation that will be used throughout this dissertation. Additional/specific notation is defined in each of the following chapters.

We adopt non-asymptotic big-oh notation: For functions $f, g : \mathcal{Z} \to \mathbb{R}_+$, we write $f = O(g)$ (resp. $f = \Omega(g)$) if there exists a constant $C > 0$ such that $f(z) \leqslant Cg(z)$ (resp. $f(z) \geqslant Cg(z)$) for all $z \in \mathcal{Z}$. We write $f = \widetilde{O}(g)$ if $f = O(g \cdot \text{polylog}(T))$, $f = \widetilde{\Omega}(g)$ if $f = \Omega(g/\text{polylog}(T))$. We use $\lesssim$ only in informal statements to highlight salient elements of an inequality.

For a vector $z \in \mathbb{R}^d$, we let $\|z\|$ denote the euclidean norm. We define $\|z\|_W^2 := \langle z, Wz \rangle$ for a positive definite matrix $W \in \mathbb{R}^{d \times d}$. For an integer $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, \dots, n\}$. For a set $\mathcal{Z}$, we let $\Delta(\mathcal{Z})$ denote the set of all Radon probability measures over $\mathcal{Z}$. We let $\text{conv}(\mathcal{Z})$ denote the set of all finitely supported convex combinations of elements in $\mathcal{Z}$. When $\mathcal{Z}$ is finite, we let $\text{unif}(\mathcal{Z})$ denote the uniform distribution over all the elements in $\mathcal{Z}$. We let $\mathbb{I}_z \in \Delta(\mathcal{Z})$ denote the delta distribution on z. We use the convention $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$.

Part I

Active Learning with Noisy Data and Rich Model Classes

The goal of active learning is to achieve the same accuracy achievable by passive learning, while using much fewer labels. Exponential savings in terms of label complexity have been proved in very special cases, but fundamental lower bounds show that such improvements are impossible in general. This suggests a need to explore alternative goals for active learning. Learning with abstention is one such alternative. In this setting, the active learning algorithm may abstain from prediction and incur an error that is marginally smaller than random guessing. We develop the first computationally efficient active learning algorithm with abstention. Our algorithm provably achieves $\mathsf{polylog}(\frac{1}{\epsilon})$ label complexity, without any low noise conditions. Such performance guarantee reduces the label complexity by an exponential factor, relative to passive learning and active learning that is not allowed to abstain. Furthermore, our algorithm is guaranteed to only abstain on hard examples (where the true label distribution is close to a fair coin), a novel property we term *proper abstention* that also leads to a host of other desirable characteristics (e.g., recovering minimax guarantees in the standard setting, and avoiding the undesirable "noise-seeking" behavior often seen in active learning). We also provide novel extensions of our algorithm that achieve *constant* label complexity and deal with model misspecification.

2.1 Introduction

Active learning aims at learning an accurate classifier with a small number of labeled data points (Settles, 2009; Hanneke, 2014). Active learning has become increasingly important in modern application of machine learning, where unlabeled data points are abundant yet the labeling process requires expensive time and effort. Empirical successes of active learning have been observed in many areas (Tong and Koller, 2001; Gal et al., 2017; Sener and Savarese, 2018). In noise-free or certain low-noise cases (i.e., under Massart noise (Massart and Nédélec, 2006)), active learning

algorithms with *provable* exponential savings over the passive counterpart have been developed (Balcan et al., 2007; Hanneke, 2007; Dasgupta et al., 2009; Hsu, 2010; Dekel et al., 2012; Hanneke, 2014; Zhang and Chaudhuri, 2014; Krishnamurthy et al., 2019; Katz-Samuels et al., 2021). On the other hand, however, not much can be said in the general case. In fact, Kääriäinen (2006) provides a $\Omega(\frac{1}{\varepsilon^2})$ lower bound by reducing active learning to a simple mean estimation problem: It takes $\Omega(\frac{1}{\varepsilon^2})$ samples to distinguish $\eta(x) = \frac{1}{2} + \varepsilon$ and $\eta(x) = \frac{1}{2} - \varepsilon$. Even with the relatively benign Tsybakov noise (Tsybakov, 2004), Castro and Nowak (2006, 2008) derive a $\Omega(\text{poly}(\frac{1}{\varepsilon}))$ lower bound, again, indicating that exponential speedup over passive learning is not possible in general. These fundamental lower bounds lay out statistical barriers to active learning, and suggests considering a refinement of the label complexity goals in active learning (Kääriäinen, 2006).

Inspecting these lower bounds, one can see that active learning suffers from classifying hard examples that are close to the decision boundary. However, *do we really require a trained classifier to do well on those hard examples?* In high-risk domains such as medical imaging, it makes more sense for the classifier to abstain from making the decision and leave the problem to a human expert. Such idea is formalized under Chow's error (Chow, 1970): Whenever the classifier chooses to abstain, a loss that is barely smaller than random guessing, i.e., $\frac{1}{2} - \gamma$, is incurred. The parameter γ should be thought as a small positive quantity, e.g., $\gamma = 0.01$. The inclusion of abstention is not only practically interesting, but also provides a statistical refinement of the label complexity goal of active learning: Achieving exponential improvement under Chow's excess error. When abstention is allowed as an action, Puchkin and Zhivotovskiy (2021) shows, for the first time, that exponential improvement in label complexity can be achieved by active learning in the general setting. However, the approach provided in Puchkin and Zhivotovskiy (2021) can not be efficiently implemented. Their algorithm follows the disagreement-based approach and requires maintaining a version space and checking whether or not an example lies in the region of disagreement. It is not clear how to generally implement these operations besides enumeration (Beygelzimer et al., 2010). Moreover, their algorithm relies on an Empirical Risk Minimization (ERM) oracle, which is known to be NP-Hard

even for a simple linear hypothesis class (Guruswami and Raghavendra, 2009).

In this chapter, we break the computational barrier and design an efficient active learning algorithm with exponential improvement in label complexity relative to conventional passive learning. The algorithm relies on weighted square loss regression oracle, which can be efficiently implemented in many cases (Krishnamurthy et al., 2017, 2019; Foster et al., 2018, 2020c). The algorithm also abstains properly, i.e., abstain only when it is the optimal choice, which allows us to easily translate the guarantees to the *standard* excess error. Along the way, we propose new noise-seeking noise conditions and show that: "uncertainty-based" active learners can be easily trapped, yet our algorithm provably overcome these noise-seeking conditions. As an extension, we also provide the first algorithm that enjoys *constant* label complexity for a *general* set of regression functions.

2.1.1 Problem Setting

Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ denote the label space. We focus on the binary classification problem where $\mathcal{Y} = \{+1, -1\}$. The joint distribution over $\mathcal{X} \times \mathcal{Y}$ is denoted as $\mathcal{D}_{\mathcal{X}\mathcal{Y}}$. We use $\mathcal{D}_{\mathcal{X}}$ to denote the marginal distribution over the input space $\mathcal{X}$, and use $\mathcal{D}_{\mathcal{Y}|x}$ to denote the conditional distribution of $\mathcal{Y}$ with respect to any $x \in \mathcal{X}$. We define $\eta(x) := \mathbb{P}_{y\sim\mathcal{D}_{\mathcal{Y}|x}}(y = +1)$ as the conditional probability of taking a positive label. We consider the standard active learning setup where $(x, y) \sim \mathcal{D}_{\mathcal{X}\mathcal{Y}}$ but y is observed only after a label querying. We consider hypothesis class $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$. For any classifier $h \in \mathcal{H}$, its (standard) error is defined as $\mathrm{err}(h) := \mathbb{P}_{(x,y)\sim\mathcal{D}_{\mathcal{X}\mathcal{Y}}}(h(x) \neq y)$.

Function approximation. We focus on the case where the hypothesis class $\mathcal{H}$ is induced from a set of regression functions $\mathcal{F} : \mathcal{X} \to [0, 1]$ that predicts the conditional probability $\eta(x)$. We write $\mathcal{H} = \mathcal{H}_{\mathcal{F}} := \{h_f : f \in \mathcal{F}\}$ where $h_f(x) := \mathrm{sign}(2f(x) - 1)$. The "size" of $\mathcal{F}$ is measured by the well-known complexity measure: the *Pseudo dimension* $\mathrm{Pdim}(\mathcal{F})$ (Pollard, 1984; Haussler, 1989, 1995). We assume $\mathrm{Pdim}(\mathcal{F}) < \infty$

throughout this chapter.[1] Following existing works in active learning (Dekel et al., 2012; Krishnamurthy et al., 2017, 2019) and contextual bandits (Agarwal et al., 2012; Foster et al., 2018; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2020), we make the following *realizability* assumption.

Assumption 2.1 (Realizability)**.** *The learner is given a set of regressors* $\mathcal{F} : \mathcal{X} \to [0, 1]$ *such that there exists a* $f^\star \in \mathcal{F}$ *characterize the true conditional probability, i.e.,* $f^\star = \eta$.

The realizability assumption allows *rich function approximation*, which strictly generalizes the setting with linear function approximation studied in active learning (e.g., in (Dekel et al., 2012)). We relax Assumption 2.1 in Section 2.4.2 to deal with model misspecification.

Regression oracle. We consider a regression oracle over $\mathcal{F}$, which is extensively studied in the literature in active learning and contextual bandits (Krishnamurthy et al., 2017, 2019; Foster et al., 2018, 2020c). Given any set $\mathcal{S}$ of weighted examples $(w, x, y) \in \mathbb{R}_+ \times \mathcal{X} \times \mathcal{Y}$ as input, the regression oracle outputs

$$\widehat{f} = \arg\min_{f \in \mathcal{F}} \sum_{(w,x,y) \in \mathcal{S}} w(f(x) - y)^2. \tag{2.1}$$

The regression oracle solves a convex optimization problem with respect to the regression function, and admits closed-form solutions in many cases, e.g., it is reduced to least squares when f is linear. We view the implementation of the regression oracle as an efficient operation and quantify the computational complexity in terms of the number of calls to the regression oracle.

Chow's excess error (Chow, 1970). Let $h^\star := h_{f^\star} \in \mathcal{H}$ denote the Bayes classifier. The *standard excess error* of classifier $h \in \mathcal{H}$ is defined as $\mathrm{err}(h) - \mathrm{err}(h^\star)$. Since achieving exponential improvement (of active over passive learning) with respect to

[1] See Section 2.5.2 for formal definition of the Pseudo dimension. Many function classes of practical interests have finite Pseudo dimension: (1) when $\mathcal{F}$ is finite, we have $\mathrm{Pdim}(\mathcal{F}) = O(\log|\mathcal{F}|)$; (2) when $\mathcal{F}$ is a set of linear functions/generalized linear function with non-decreasing link function, we have $\mathcal{F} = O(d)$; (3) when $\mathcal{F}$ is a set of degree-r polynomial in $\mathbb{R}^d$, we have $\mathrm{Pdim}(\mathcal{F}) = O(\binom{d+r}{r})$.

the standard excess error is impossible in general (Kääriäinen, 2006), we introduce Chow's excess error next. We consider classifier of the form $\widehat{h} : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ where $\perp$ denotes the action of abstention. For any fixed $0 < \gamma < \frac{1}{2}$, the Chow's error is defined as

$$\mathrm{err}_\gamma(\widehat{h}) := \mathbb{P}_{(x,y)\sim\mathcal{D}_{xy}}(\widehat{h}(x) \neq y, \widehat{h}(x) \neq \perp) + (1/2 - \gamma) \cdot \mathbb{P}_{(x,y)\sim\mathcal{D}_{xy}}(\widehat{h}(x) = \perp). \tag{2.2}$$

The parameter γ can be chosen as a small constant, e.g., $\gamma = 0.01$, to avoid excessive abstention: The price of abstention is only marginally smaller than random guess. The *Chow's excess error* is then defined as $\mathrm{err}_\gamma(\widehat{h}) - \mathrm{err}(h^\star)$ (Puchkin and Zhivotovskiy, 2021). For any fixed accuracy level $\varepsilon > 0$, we aim at constructing a classifier $\widehat{h} : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ with ε Chow's excess error and polylog$(\frac{1}{\varepsilon})$ label complexity. We also relate Chow's excess error to standard excess error in Section 2.3.

2.1.2 Why Learning with Chow's Excess Error Helps?

For illustration purpose, we focus on the simple case where $\mathcal{X} = \{x\}$ in this section. The active learning problem is then reduced to mean estimation of the conditional probability $\eta(x) \in [0, 1]$.

Learning with standard excess error. Fix any $\varepsilon > 0$. With respect to the conditional probability $\eta(x)$, we define the positive region $\mathcal{S}_{+,\varepsilon} := \left[\frac{1-\varepsilon}{2}, 1\right]$ and the negative region $\mathcal{S}_{-,\varepsilon} := [0, \frac{1+\varepsilon}{2}]$. These regions are interpreted as follows: If $\eta(x) \in \mathcal{S}_{+,\varepsilon}$ (resp. $\eta(x) \in \mathcal{S}_{-,\varepsilon}$), then labeling x as positive (resp. negative) incurs no more than ε excess error. Under standard excess error, we define the flexible region as $\mathcal{S}^{\text{standard}}_{\text{flexible},\varepsilon} := \mathcal{S}_{+,\varepsilon} \cap \mathcal{S}_{-,\varepsilon} = [\frac{1-\varepsilon}{2}, \frac{1+\varepsilon}{2}]$ (the grey area in the top plot in Fig. 2.1). Two important implications of the flexible region are as follows: (1) if $\eta(x) \in \mathcal{S}^{\text{standard}}_{\text{flexible},\varepsilon}$, labeling x as either positive or negative would lead to excess error at most ε; and (2) if $\eta(x) \notin \mathcal{S}^{\text{standard}}_{\text{flexible},\varepsilon}$, then a classifier must correctly label x as either positive or negative to guarantee ε excess error. Since the flexible region is of length ε under

standard excess error, distinguishing two points at the edge of the flexible region, e.g., $\eta(x) = \frac{1}{2} - \varepsilon$ and $\eta(x) = \frac{1}{2} + \varepsilon$, leads to label complexity lower bound $\Omega(\frac{1}{\varepsilon^2})$.

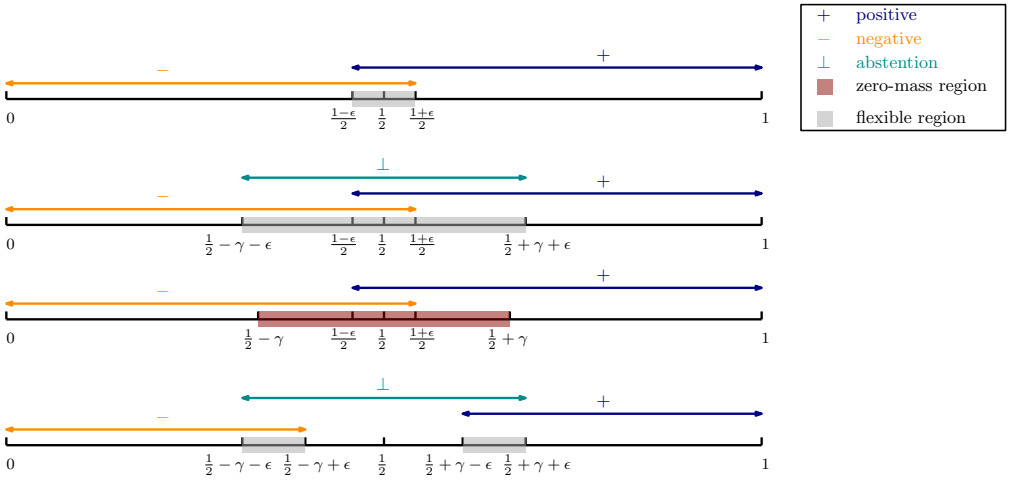


Figure 2.1: Illustration of regions $\mathcal{S}_{+,\varepsilon}$, $\mathcal{S}_{-,\varepsilon}$ and $\mathcal{S}_{\perp,\varepsilon}$. Top row: Learning with standard excess error $\mathrm{err}(\widehat{h}) - \mathrm{err}(h^\star)$. Second row: Learning with standard excess error $\mathrm{err}(\widehat{h}) - \mathrm{err}(h^\star)$ and Massart's noise with parameter γ. Third row: Learning with Chow's excess error $\mathrm{err}_\gamma(\widehat{h}) - \mathrm{err}(h^\star)$. Bottom row: Learning against the optimal Chow's excess error, i.e., $\mathrm{err}_\gamma(\widehat{h}) - \inf_{h:\mathcal{X}\to\{+1,-1,\perp\}} \mathrm{err}_\gamma(h)$.

Learning with Chow's excess error. Now we turn our attention to the Chow's error. We consider $\mathcal{S}_{+,\varepsilon}$ and $\mathcal{S}_{-,\varepsilon}$ as before, and define a third abstention region $\mathcal{S}_{\perp,\varepsilon} := [\frac{1}{2}-\gamma-\varepsilon, \frac{1}{2}+\gamma+\varepsilon]$ If $\eta(x) \in \mathcal{S}_{\perp,\varepsilon}$, then abstaining on x leads to Chow's excess error at most ε. We define the positive flexible region as $\mathcal{S}^{\mathrm{Chow}}_{\mathrm{flexible},+,\varepsilon} := \mathcal{S}_{\perp,\varepsilon} \cap \mathcal{S}_{\perp,\varepsilon} = [\frac{1-\varepsilon}{2}, \frac{1}{2}+\gamma+\varepsilon]$, which is of length $\gamma + \frac{3\varepsilon}{2}$. The negative flexible region is defined similarly (see the second plot in Fig. 2.1). Since both positive and negative flexible regions are longer than γ, it takes at most $\widetilde{O}(\frac{1}{\gamma^2})$ samples to identify a labeling action with at most ε Chow's excess error. Without loss of generality, we assume $\eta(x) < \frac{1}{2}$. It takes $\widetilde{O}(\frac{1}{\gamma^2})$ samples to construct a confidence interval $[\mathrm{lcb}(x), \mathrm{ucb}(x)]$ (of $\eta(x)$) of length at most $\frac{\gamma}{2}$. If $\eta(x) < \frac{1-\gamma}{2}$, we observe $\mathrm{ucb}(x) \leqslant \frac{1}{2}$ and know that labeling x negative ensures at most ε excess error. If $\eta(x) \in [\frac{1-\gamma}{2}, \frac{1}{2}]$, we observe $[\mathrm{lcb}(x), \mathrm{ucb}(x)] \subseteq [\frac{1}{2}-\gamma, \frac{1}{2}+\gamma] \subseteq \mathcal{S}_{\perp,\varepsilon}$. Thus, choosing to abstain on x again ensures at

most ε excess error. To summarize, learning with Chow's excess error in the general case resembles the behavior of learning under Massart noise: Hard examples that are close to the decision boundary are not selected for labeling (see the third plot in Fig. 2.1), and learning is possible with $\widetilde{O}(\frac{1}{\gamma^2})$ labels.

Why not compete against the optimal Chow's error. As shown in the last plot in Fig. 2.1, the flexible regions become narrow (of length $O(\varepsilon)$) again when competing against the optimal Chow's error. Abstention becomes the only action that guarantees at most ε excess error over region $(\frac{1}{2}-\gamma-\varepsilon, \frac{1}{2}+\gamma+\varepsilon)$. One then needs to distinguish cases $\eta(x) = \frac{1}{2}+\gamma-2\varepsilon$ and $\eta(x) = \frac{1}{2}+\gamma+2\varepsilon$, which requires $\Omega(\frac{1}{\varepsilon^2})$ samples. It is also unreasonable to compete against the optimal Chow's error: When $\eta(x) = \frac{1}{2}+\gamma-2\varepsilon$, $\Omega(\frac{1}{\varepsilon^2})$ samples are required to decide whether take action $+1$ or action $\perp$; however, with only $\widetilde{O}(\frac{1}{\gamma^2})$ samples, one can already determine $\eta(x) > \frac{1}{2}$.

2.1.3 Contributions and Organization

We provide informal statements of our main results in this section. Our results depend on complexity measures such as *value function* disagreement coefficient θ and eluder dimension $\mathfrak{e}$ (formally defined in Section 2.2 and Section 2.5.1). These complexity measures are previously analyzed in contextual bandits (Russo and Van Roy, 2013; Foster et al., 2020c) and we import them to the active learning setup. These complexity measures are well-bounded for many function classes of practical interests, e.g., we have $\theta, \mathfrak{e} = \widetilde{O}(d)$ for linear and generalized linear functions on $\mathbb{R}^d$.

Our first main contribution is that we design the first *computationally efficient* active learning algorithm (Algorithm 1) that achieves exponential labeling savings, *without any low noise assumptions*.

Theorem 2.2 (Informal)**.** *There exists an algorithm that constructs a classifier* $\widehat{h} : \mathcal{X} \to \{+1, -1, \perp\}$ *with Chow's excess error at most* ε *and label complexity* $\widetilde{O}(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\gamma^2} \cdot \operatorname{polylog}(\frac{1}{\varepsilon}))$*, without any low noise assumptions. The algorithm can be efficiently im-*

plemented via a regression oracle: It takes $\widetilde{O}(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma^3})$ *oracle calls for general* $\mathcal{F}$*, and* $\widetilde{O}(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma})$ *oracle calls for convex* $\mathcal{F}$*.*

The formal statements are provided in Section 2.2. The *statistical* guarantees (i.e., label complexity) in Theorem 2.2 is similar to the one achieved in Puchkin and Zhivotovskiy (2021), with one critical difference: The label complexity provided in Puchkin and Zhivotovskiy (2021) is in terms of the *classifier-based* disagreement coefficient $\check{\theta}$ (Hanneke, 2014). Even for a set of linear classifier, $\check{\theta}$ is only known to be bounded in special cases, e.g., when $\mathcal{D}_X$ is uniform over the unit sphere (Hanneke, 2007). On the other hand, we have $\theta \leqslant d$ for any $\mathcal{D}_X$ (Foster et al., 2020c).

We say that a classifier $\widehat{h} : \mathcal{X} \to \{+1, -1, \perp\}$ enjoys proper abstention if it abstains only if abstention is indeed the optimal choice (based on Eq. (2.2)). For any classifier that enjoys proper abstention, one can easily relate its *standard* excess error to the Chow's excess error, under commonly studied Massart/Tsybakov noises (Massart and Nédélec, 2006; Tsybakov, 2004). The classifier obtained in Theorem 2.2 enjoys proper abstention, and achieves the following guarantees (formally stated in Section 2.3.1).

Theorem 2.3 (Informal)**.** *Under Massart/Tsybakov noise, with appropriate adjustments, the classifier learned in Theorem 2.2 achieves the minimax optimal label complexity under standard excess error.*

We also propose new noise conditions that *strictly* generalize the usual Massart/Tsybakov noises, which we call noise-seeking conditions. At a high-level, the noise-seeking conditions allow abundant data points with $\eta(x)$ equal/close to $\frac{1}{2}$. These points are somewhat "harmless" since it hardly matters what label is predicted at that point (in terms of excess error). These seemingly "harmless" data points can, however, cause troubles for any active learning algorithm that requests the label for any point that is uncertain, i.e., the algorithm cannot decide if $|\eta(x) - \frac{1}{2}|$ is strictly greater than 0. We call such algorithms "uncertainty-based" active learners. These algorithms could wastefully sample in these "harmless" regions, ignoring other regions where erring could be much more harmful. We

derive the following proposition (formally stated in Section 2.3.2) under these noise-seeking conditions.

Proposition 2.4 (Informal). *For any labeling budget* $B \gtrsim \frac{1}{\gamma^2} \cdot \mathrm{polylog}(\frac{1}{\varepsilon})$, *there exists a learning problem such that* (1) *any uncertainty-based active learner suffers standard excess error* $\Omega(B^{-1})$; *yet* (2) *the classifier* $\widehat{h}$ *learned in Theorem 2.2 achieves standard excess error at most* ε.

The above result demonstrates the superiority of our algorithm over any "uncertainty-based" active learner. Moreover, we show that, under these strictly harder noise-seeking conditions, our algorithm still achieve guarantees similar to the ones stated in Theorem 2.3.

Before presenting our next main result, we first consider a simple active learning problem with $\mathcal{X} = \{x\}$. Under Massart noise, we have $|\eta(x) - \frac{1}{2}| \geqslant \tau_0$ for some constant $\tau_0 > 0$. Thus, it takes no more than $O(\tau_0^{-2} \log \frac{1}{\delta})$ labels to achieve ε standard excess error, no matter how small ε is. This example shows that, at least in simple cases, we can expect to achieve a *constant* label complexity for active learning, with no dependence on $\frac{1}{\varepsilon}$ at all. To the best of our knowledge, our next result provides the first generalization of such phenomenon to a *general* set of (finite) regression functions, as long as its eluder dimension $\mathfrak{e}$ is bounded.

Theorem 2.5 (Informal). *Under Massart noise with parameter* τ_0 *and a general* (*finite*) *set of regression function* $\mathcal{F}$. *There exists an algorithm that returns a classifier with standard excess error at most* ε *and label complexity* $O(\frac{\mathfrak{e} \cdot \log(|\mathcal{F}|/\delta)}{\tau_0^2})$, *which is independent of* $\frac{1}{\varepsilon}$.

A similar constant label complexity holds with Chow's excess error, without any low noise assumptions. We also provide discussion on why previous algorithms do not achieve such constant label complexity, even in the case with linear functions. We defer formal statements and discussion to Section 2.4.1. In Section 2.4.2, we relax Assumption 2.1 and propose an algorithm that can deal with model misspecification.

Organization. The rest of this chapter is organized as follows. We discuss additional related work in Section 2.1.4. We present our main algorithm and its guarantees in Section 2.2. We further analyze our algorithm under standard excess error in Section 2.3, and discuss other important properties of the algorithm. Extensions of our algorithm, e.g., achieving *constant* label complexity and dealing with model misspecification, are provided in Section 2.4. We defer all proofs to Section 2.5.

2.1.4 Additional Related Work

Learning with Chow's excess error is closely related to learning under Massart noise (Massart and Nédélec, 2006), which assumes that no data point has conditional expectation close to the decision boundary, i.e., $\mathbb{P}(|\eta(x) - 1/2| \leqslant \tau_0) = 0$ for some constant $\tau_0 > 0$. Learning with Massart noise is commonly studied in active learning (Balcan et al., 2007; Hanneke, 2014; Zhang and Chaudhuri, 2014; Krishnamurthy et al., 2019), where $\widetilde{O}(\tau_0^{-2})$ type of guarantees are achieved. Instead of making explicit assumptions on the underlying distribution, learning with Chow's excess error empowers the learner with the ability to abstain: There is no need to make predictions on hard data points that are close to the decision boundary, i.e., $\{x : |\eta(x) - 1/2| \leqslant \gamma\}$. Learning with Chow's excess error thus works on more general settings and still enjoys the $\widetilde{O}(\gamma^{-2})$ type of guarantee as learning under Massart noise (Puchkin and Zhivotovskiy, 2021).[2] We show in Section 2.3 that statistical guarantees achieved under Chow's excess error can be directly translated to guarantees under (usual and more challenging versions of) Massart/Tsybakov noise (Massart and Nédélec, 2006; Tsybakov, 2004).

Active learning at aim competing the best in-class classifier with few labels. A long line of work directly works with the set of classifiers (Balcan et al., 2007; Hanneke, 2007, 2014; Huang et al., 2015; Puchkin and Zhivotovskiy, 2021), where the algorithms are developed with (in general) hard-to-implement ERM oracles

[2] However, passive learning with abstention only achieves error rate $\frac{1}{n\gamma}$ with n samples (Bousquet and Zhivotovskiy, 2021).

(Guruswami and Raghavendra, 2009) and the the guarantees dependence on the so-called disagreement coefficient (Hanneke, 2014). More recently, learning with function approximation have been studied inactive learning and contextual bandits (Dekel et al., 2012; Agarwal et al., 2012; Foster et al., 2018; Krishnamurthy et al., 2019). The function approximation scheme permits efficient regression oracles, which solve convex optimization problems with respect to regression functions (Krishnamurthy et al., 2017, 2019; Foster et al., 2018). It can also be analyzed with the scale-sensitive version of disagreement coefficient, which is usually tighter than the original one (Foster et al., 2020c; Russo and Van Roy, 2013). Our algorithms are inspired Krishnamurthy et al. (2019), where the authors study active learning under the standard excess error. The main deviation from Krishnamurthy et al. (2019) is that we need to *manually* construct a classifier $\widehat{h}$ with an abstention option and $\widehat{h} \notin \mathcal{H}$, which leads to differences in the analysis of excess error and label complexity. We borrow techniques developed in contextual bandits Russo and Van Roy (2013); Foster et al. (2020c) to analyze our algorithm.

Although one can also apply our algorithms in the nonparametric regime with proper pre-processing schemes such discretizations, our algorithm primarily works in the parametric setting with finite pseudo dimension (Haussler, 1995) and finite (value function) disagreement coefficient (Foster et al., 2020c). Active learning has also been studied in the nonparametric regime (Castro and Nowak, 2008; Koltchinskii, 2010; Minsker, 2012; Locatelli et al., 2017). Notably, Shekhar et al. (2021) studies Chow's excess error with margin-type of assumptions. Their setting is different to ours and poly$(\frac{1}{\varepsilon})$ label complexities are achieved. If abundant amounts of data points are allowed to be exactly at the decision boundary, i.e., $\eta(x) = \frac{1}{2}$, Kpotufe et al. (2021) recently shows that, in the nonparametric regime, no active learner can outperform the passive counterpart.

2.2 Efficient Active Learning with Abstention

We provide our main algorithm (Algorithm 1) in this section. Algorithm 1 is an adaptation of the algorithm developed in Krishnamurthy et al. (2017, 2019), which

studies active learning under the standard excess error (and Massart/Tsybakov noises). We additionally take the abstention option into consideration, and *manually construct* classifiers using the active set of (uneliminated) regression functions (which do not belong to the original hypothesis class). These new elements allow us to achieve ε Chow's excess error with polylog$(\frac{1}{\varepsilon})$ label complexity, without any low noise assumptions.

Algorithm 1 Efficient Active Learning with Abstention

Input: Accuracy level $\varepsilon > 0$, abstention parameter $\gamma \in (0, 1/2)$ and confidence level $\delta \in (0, 1)$.

Define $T := \widetilde{O}(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma})$, $M := \lceil \log_2 T \rceil$ and $C_\delta := O(\operatorname{Pdim}(\mathcal{F}) \cdot \log(T/\delta))$.

Define $\tau_m := 2^m$ for $m \geqslant 1$, $\tau_0 := 0$ and $\beta_m := (M - m + 1) \cdot C_\delta$.

for epoch $m = 1, 2, \ldots, M$ **do**

  Get $\widehat{f}_m := \arg\min_{f \in \mathcal{F}} \sum_{t=1}^{\tau_{m-1}} Q_t (f(x_t) - y_t)^2$.

    `//We use` $Q_t \in \{0, 1\}$ `to indicate whether the label of` x_t `is queried.`

  (Implicitly) Construct active set of regression functions $\mathcal{F}_m \subseteq \mathcal{F}$ as

$$\mathcal{F}_m := \left\{ f \in \mathcal{F} : \sum_{t=1}^{\tau_{m-1}} Q_t (f(x_t) - y_t)^2 \leqslant \sum_{t=1}^{\tau_{m-1}} Q_t (\widehat{f}_m(x_t) - y_t)^2 + \beta_m \right\}.$$

  Construct classifier $\widehat{h}_m : \mathcal{X} \to \{+1, -1, \perp\}$ as

$$\widehat{h}_m(x) := \begin{cases} \perp, & \text{if } [\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)] \subseteq \left[\frac{1}{2} - \gamma, \frac{1}{2} + \gamma\right]; \\ \operatorname{sign}(2\widehat{f}_m(x) - 1), & \text{o.w.} \end{cases}$$

  and construct query function $g_m(x) := \mathbb{1}\left(\frac{1}{2} \in (\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m))\right) \cdot \mathbb{1}(\widehat{h}_m(x) \neq \perp)$.

  if epoch $m = M$ **then**

    Return classifier $\widehat{h}_M$.

  for time $t = \tau_{m-1} + 1, \ldots, \tau_m$ **do**

    Observe $x_t \sim \mathcal{D}_X$. Set $Q_t := g_m(x_t)$.

    if $Q_t = 1$ **then**

      Query the label y_t of x_t.

Algorithm 1 runs in epochs of geometrically increasing lengths. At the begin-

ning of epoch $m \in [M]$, Algorithm 1 first computes the empirical best regression function $\widehat{f}_m$ that achieves the smallest cumulative square loss over previously labeled data points ($\widehat{f}_1$ can be selected arbitrarily); it then (implicitly) constructs an active set of regression functions $\mathcal{F}_m$, where the cumulative square loss of each $f \in \mathcal{F}_m$ is not too much larger than the cumulative square loss of empirical best regression function $\widehat{f}_m$. For any $x \in \mathcal{X}$, based on the active set of regression functions, Algorithm 1 constructs a lower bound $\mathsf{lcb}(x; \mathcal{F}_m) := \inf_{f \in \mathcal{F}_m} f(x)$ and an upper bound $\mathsf{ucb}(x; \mathcal{F}_m) := \sup_{f \in \mathcal{F}_m} f(x)$ for the true conditional probability $\eta(x)$. An empirical classifier $\widehat{h}_m : \mathcal{X} \to \{+1, -1, \perp\}$ and a query function $g_m : \mathcal{X} \to \{0, 1\}$ are then constructed based on these confidence ranges and the abstention parameter γ. For any time step t within epoch m, Algorithm 1 queries the label of the observed data point x_t if and only if $Q_t := g_m(x_t) = 1$. Algorithm 1 returns $\widehat{h}_M$ as the learned classifier.

We now discuss the empirical classifier $\widehat{h}_m$ and the query function g_m in more detail. Consider the event where $f^\star \in \mathcal{F}_m$ for all $m \in [M]$, which can be shown to hold with high probability. The constructed confidence intervals are valid under this event, i.e., $\eta(x) \in [\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)]$. First, let us examine the conditions that determine a label query. The label of x is *not* queried if

- **Case 1:** $\widehat{h}_m(x) = \perp$**.** We have $\eta(x) \in [\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)] \subseteq [\frac{1}{2} - \gamma, \frac{1}{2} + \gamma]$. Abstention leads to the smallest error (Herbei and Wegkamp, 2006), and no query is needed.

- **Case 2:** $\frac{1}{2} \notin (\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m))$**.** We have $\mathsf{sign}(2\widehat{f}_m(x) - 1) = \mathsf{sign}(2f^\star(x) - 1)$. Thus, no excess error is incurred and there is no need to query.

The only case when label query *is* issued, and thus when the classifier $\widehat{h}_m$ may suffer from excess error, is when

$$\frac{1}{2} \in (\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)) \quad \text{and} \quad [\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)] \nsubseteq \left[\frac{1}{2} - \gamma, \frac{1}{2} + \gamma\right] \tag{2.3}$$

hold simultaneously. Eq. (2.3) necessarily leads to the condition $w(x; \mathcal{F}_m) := \mathsf{ucb}(x; \mathcal{F}_m) - \mathsf{lcb}(x; \mathcal{F}_m) > \gamma$. Our theoretical analysis shows that the event must $\mathbb{1}(w(x; \mathcal{F}_m) > \gamma)$ happens infrequently, and its frequency is closely related to the so-called *value function disagreement coefficient* (Foster et al., 2020c), which we introduce as follows.[3]

Definition 2.6 (Value function disagreement coefficient)**.** *For any* $f^\star \in \mathcal{F}$ *and* $\gamma_0, \varepsilon_0 > 0$*, the value function disagreement coefficient* $\theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0)$ *is defined as*

$$\sup_{\mathcal{D}_X} \sup_{\gamma > \gamma_0, \varepsilon > \varepsilon_0} \left\{ \frac{\gamma^2}{\varepsilon^2} \cdot \mathbb{P}_{\mathcal{D}_X}\left(\exists f \in \mathcal{F} : |f(x) - f^\star(x)| > \gamma, \|f - f^\star\|_{\mathcal{D}_X} \leqslant \varepsilon\right) \right\} \vee 1,$$

where $\|f\|^2_{\mathcal{D}_X} := \mathbb{E}_{x \sim \mathcal{D}_X}[f^2(x)]$.

Combining the insights discussed above, we derive the following label complexity guarantee for Algorithm 1 (we use $\theta := \sup_{f^\star \in \mathcal{F}, \iota > 0} \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma/2, \iota)$ and discuss its boundedness below). [4]

Theorem 2.7. *With probability at least* $1-2\delta$*, Algorithm 1 returns a classifier with Chow's excess error at most* ε *and label complexity* $O(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\gamma^2} \cdot \log^2(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma}) \cdot \log(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma \delta}))$.

Theorem 2.7 shows that Algorithm 1 achieves exponential label savings (i.e., $\operatorname{polylog}(\frac{1}{\varepsilon})$) without any low noise assumptions. We discuss the result in more detail next.

- **Boundedness of** θ**.** The value function disagreement coefficient is well-bounded for many function classes of practical interests. For instance, we have $\theta \leqslant d$ for linear functions on $\mathbb{R}^d$ and $\theta \leqslant C_{\mathsf{link}} \cdot d$ for generalized linear functions (where C_{link} is a quantity related to the link function). Moreover,

[3] Compared to the original definition studied in contextual bandits (Foster et al., 2020c), our definition takes an additional "sup" over all possible marginal distributions $\mathcal{D}_X$ to account for *distributional shifts* incurred by selective querying (which do not occur in contextual bandits). Nevertheless, as we show below, our disagreement coefficient is still well-bounded for many important function classes.

[4] It suffices to take $\theta := \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma/2, \iota)$ with $\iota \propto \sqrt{\gamma\varepsilon}$ to derive a slightly different guarantee. See Section 2.5.3.

θ is *always* upper bounded by complexity measures such as (squared) star number and eluder dimension (Foster et al., 2020c). See Section 2.5.1 for the detailed definitions/bounds.

- **Comparison to Puchkin and Zhivotovskiy (2021).** The label complexity bound derived in Theorem 2.7 is similar to the one derived in Puchkin and Zhivotovskiy (2021), with one critical difference: The bound derived in Puchkin and Zhivotovskiy (2021) is in terms of *classifier-based* disagreement coefficient $\check{\theta}$ (Hanneke, 2014). Even in the case with linear classifiers, $\check{\theta}$ is only known to be bounded under additional assumptions, e.g., when $\mathcal{D}_X$ is uniform over the unit sphere.

Computational efficiency. We discuss how to efficiently implement Algorithm 1 with the regression oracle defined in Eq. (2.1). [5] Our implementation relies on subroutines developed in Krishnamurthy et al. (2017); Foster et al. (2018), which allow us to approximate confidence bounds $\mathsf{ucb}(x;\mathcal{F}_m)$ and $\mathsf{lcb}(x;\mathcal{F}_m)$ up to α approximation error with $O(\frac{1}{\alpha^2}\log\frac{1}{\alpha})$ (or $O(\log\frac{1}{\alpha})$ when $\mathcal{F}$ is convex and closed under pointwise convergence) calls to the regression oracle. To achieve the same theoretical guarantees shown in Theorem 2.7 (up to changes in constant terms), we show that it suffices to (i) control the approximation error at level $O(\frac{\gamma}{\log T})$, (ii) construct the approximated confidence bounds $\widehat{\mathsf{lcb}}(x;\mathcal{F}_m)$ and $\widehat{\mathsf{ucb}}(x;\mathcal{F}_m)$ in a way such that the confidence region is non-increasing with respect to the epoch m, i.e., $(\widehat{\mathsf{lcb}}(x;\mathcal{F}_m),\widehat{\mathsf{ucb}}(x;\mathcal{F}_m)) \subseteq (\widehat{\mathsf{lcb}}(x;\mathcal{F}_{m-1}),\widehat{\mathsf{ucb}}(x;\mathcal{F}_{m-1}))$ (this ensures that the sampling region is non-increasing even with *approximated* confidence bounds, which is important to our theoretical analysis), and (iii) use the approximated confidence bounds $\widehat{\mathsf{lcb}}(x;\mathcal{F}_m)$ and $\widehat{\mathsf{ucb}}(x;\mathcal{F}_m)$ to construct the classifier $\widehat{h}_m$ and the query function g_m. We provide our guarantees as follows, and leave details

[5] Recall that the implementation of the regression oracle should be viewed as an efficient operation since it solves a convex optimization problem with respect to the regression function, and it even admits closed-form solutions in many cases, e.g., it is reduced to least squares when f is linear. On the other hand, the ERM oracle used in Puchkin and Zhivotovskiy (2021) is NP-hard even for a set of linear classifiers (Guruswami and Raghavendra, 2009).

to Section 2.5.3 (we redefine $\theta := \sup_{f^\star \in \mathcal{F}, \iota > 0} \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma/4, \iota)$ in the Theorem 2.8 to account to approximation error).

Theorem 2.8. *Algorithm 1 can be efficiently implemented via the regression oracle and enjoys the same theoretical guarantees stated in Theorem 2.7. The number of oracle calls needed is* $\widetilde{O}(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma^3})$ *for a general set of regression functions* $\mathcal{F}$*, and* $\widetilde{O}(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma})$ *when* $\mathcal{F}$ *is convex and closed under pointwise convergence. The per-example inference time of the learned* $\widehat{h}_M$ *is* $\widetilde{O}(\frac{1}{\gamma^2} \log^2(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon}))$ *for general* $\mathcal{F}$*, and* $\widetilde{O}(\log \frac{1}{\gamma})$ *when* $\mathcal{F}$ *is convex and closed under pointwise convergence.*

With Theorem 2.8, we provide the first computationally efficient active learning algorithm that achieves exponential label savings, without any low noise assumptions.

2.3 Guarantees under Standard Excess Error

We provide guarantees for Algorithm 1 under *standard* excess error. In Section 2.3.1, we show that Algorithm 1 can be used to recover the usual minimax label complexity under Massart/Tsybakov noise; we also provide a new learning paradigm based on Algorithm 1 under limited budget. In Section 2.3.2, we show that Algorithm 1 provably avoid the undesired *noise-seeking* behavior often seen in active learning.

2.3.1 Recovering Minimax Optimal Label Complexity

One way to convert an abstaining classifier $\widehat{h} : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ into a standard classifier $\check{h} : \mathcal{X} \to \mathcal{Y}$ is by randomizing the prediction in its abstention region, i.e., if $\widehat{h}(x) = \perp$, then its randomized version $\check{h}(x)$ predicts $+1/-1$ with equal probability (Puchkin and Zhivotovskiy, 2021). With such randomization, the *standard excess error* of $\check{h}$ can be characterized as

$$\operatorname{err}(\check{h}) - \operatorname{err}(h^\star) = \operatorname{err}_\gamma(\widehat{h}) - \operatorname{err}(h^\star) + \gamma \cdot \mathbb{P}_{x \sim \mathcal{D}_\mathcal{X}}(\widehat{h}(x) = \perp). \tag{2.4}$$

The standard excess error depends on the (random) abstention region of $\widehat{h}$, which is difficult to quantify in general. To give a more practical characterization of the standard excess error, we introduce the concept of proper abstention in the following.

Definition 2.9 (Proper abstention)**.** *A classifier $\widehat{h} : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ enjoys proper abstention if and only if it abstains in regions where abstention is indeed the optimal choice, i.e.,* $\{x \in \mathcal{X} : \widehat{h}(x) = \perp\} \subseteq \{x \in \mathcal{X} : \eta(x) \in [\frac{1}{2} - \gamma, \frac{1}{2} + \gamma]\} =: \mathcal{X}_\gamma$.

Proposition 2.10. *The classifier $\widehat{h}$ returned by Algorithm 1 enjoys proper abstention. With randomization over the abstention region, we have the following upper bound on its standard excess error*

$$\operatorname{err}(\check{h}) - \operatorname{err}(h^\star) \leqslant \operatorname{err}_\gamma(\widehat{h}) - \operatorname{err}(h^\star) + \gamma \cdot \mathbb{P}_{x \sim \mathcal{D}_\mathcal{X}}(x \in \mathcal{X}_\gamma). \tag{2.5}$$

The proper abstention property of $\widehat{h}$ returned by Algorithm 1 is achieved via conservation: $\widehat{h}$ will avoid abstention unless it is absolutely sure that abstention is the optimal choice.[6] To characterize the standard excess error of classifier with proper abstention, we only need to upper bound the term $\mathbb{P}_{x \sim \mathcal{D}_\mathcal{X}}(x \in \mathcal{X}_\gamma)$, which does *not* depends on the (random) classifier $\widehat{h}$. Instead, it only depends on the marginal distribution. We next introduce the common Massart/Tsybakov noise conditions.

Definition 2.11 (Massart noise, Massart and Nédélec (2006))**.** *The marginal distribution $\mathcal{D}_\mathcal{X}$ satisfies the Massart noise condition with parameter $\tau_0 > 0$ if* $\mathbb{P}(x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \tau_0) = 0$.

Definition 2.12 (Tsybakov noise, Tsybakov (2004))**.** *The marginal distribution $\mathcal{D}_\mathcal{X}$ satisfies the Tsybakov noise condition with parameter $\beta \geqslant 0$ and a universal constant $c > 0$ if $\mathbb{P}(x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \tau) \leqslant c\,\tau^\beta$ for any $\tau > 0$.*

[6] On the other hand, however, the algorithm provided in Puchkin and Zhivotovskiy (2021) is very unlikely to have such property. In fact, only a small but *nonzero* upper bound of abstention rate is provided (Proposition 3.6 therein) under the Massart noise with $\gamma \leqslant \frac{\tau_0}{2}$; yet any classifier that enjoys proper abstention should have exactly zero abstention rate.

As in Balcan et al. (2007); Hanneke (2014), we assume knowledge of noise parameters (e.g., τ_0, β). Together with the active learning lower established in Castro and Nowak (2006, 2008), and focusing on the dependence of ε, our next theorem shows that Algorithm 1 can be used to recover the minimax label complexity in active learning, under the *standard* excess error.

Theorem 2.13. *With an appropriate choice of the abstention parameter γ in Algorithm 1 and randomization over the abstention region, Algorithm 1 learns a classifier $\check{h}$ at the minimax optimal rates: To achieve ε standard excess error, it takes $\widetilde{\Theta}(\tau_0^{-2})$ labels under Massart noise and takes $\widetilde{\Theta}(\varepsilon^{-2/(1+\beta)})$ labels under Tsybakov noise.*

Remark 2.14. *In addition to recovering the minimax rates, the proper abstention property is desirable in practice: It guarantees that $\widehat{h}$ will not abstain on easy examples, i.e., it will not mistakenly flag easy examples as "hard-to-classify", thus eliminating unnecessary human labeling efforts.*

Algorithm 1 can also be used to provide new learning paradigms in the limited budget setting, which we introduce below. No prior knowledge of noise parameters are required in this setup.

New learning paradigm under limited budget. Given any labeling budget $B > 0$, we can then choose $\gamma \approx B^{-1/2}$ in Algorithm 1 to make sure the label complexity is never greater than B (with high probability). The learned classifier enjoys Chow's excess error (with parameter γ) at most ε; its standard excess error (with randomization over the abstention region) can be analyzed by relating the $\gamma \cdot \mathbb{P}_{x \sim \mathcal{D}_X}(x \in \mathcal{X}_\gamma)$ term in Eq. (2.5) to the Massart/Tsybakov noise conditions, as discussed above.

2.3.2 Abstention to Avoid Noise-Seeking

Active learning algorithms sometimes exhibit *noise-seeking* behaviors, i.e., oversampling in regions where $\eta(x)$ is close to the $\frac{1}{2}$ level. Such noise-seeking behavior is known to be a fundamental barrier to achieve low label complexity (under standard

excess error), e.g., see Kääriäinen (2006). We show in this section that abstention naturally helps avoiding noise-seeking behaviors and speeding up active learning.

To better illustrate how properly abstaining classifiers avoid noise-seeking behavior, we first propose new noise conditions below, which strictly generalize the usual Massart/Tsybakov noises.

Definition 2.15 (Noise-seeking Massart noise)**.** *The marginal distribution $\mathcal{D}_X$ satisfies the noise-seeking Massart noise condition with parameters $0 \leqslant \zeta_0 < \tau_0 \leqslant 1/2$ if $\mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X} : \zeta_0 < |\eta(x) - 1/2| \leqslant \tau_0) = 0$.*

Definition 2.16 (Noise-seeking Tsybakov noise)**.** *The marginal distribution $\mathcal{D}_X$ satisfies the noise-seeking Tsybakov noise condition with parameters $0 \leqslant \zeta_0 < 1/2$, $\beta \geqslant 0$ and a universal constant $c > 0$ if $\mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X} : \zeta_0 < |\eta(x) - 1/2| \leqslant \tau) \leqslant c\,\tau^\beta$ for any $\tau > \zeta_0$.*

Compared to the standard Massart/Tsybakov noises, these newly proposed noise-seeking conditions allow arbitrary probability mass of data points whose conditional probability $\eta(x)$ is equal/close to $1/2$. As a result, they can trick standard active learning algorithms into exhibiting the noise-seeking bahaviors (and hence their names). We also mention that the parameter ζ_0 should be considered as an *extremely small quantity* (e.g., $\zeta_0 \ll \varepsilon$), with the extreme case corresponding to $\zeta_0 = 0$ (which still allow arbitrary probability for region $\{x \in \mathcal{X} : \eta(x) = 1/2\}$).

Ideally, any active learning algorithm should not be heavily affected by these noise conditions since it hardly matters (in terms of excess error) what label is predicted over region $\{x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \zeta_0\}$. However, these seemingly benign noise-seeking conditions can cause troubles for any "uncertainty-based" active learner, i.e., any active learning algorithm that requests the label for any point that is uncertain (see Definition 2.41 in Section 2.5.4 for formal definition). In particular, under limited budget, we derive the following result.

Proposition 2.17. *Fix $\varepsilon, \delta, \gamma > 0$. For any labeling budget $B \gtrsim \frac{1}{\gamma^2} \cdot \log^2(\frac{1}{\varepsilon\,\gamma}) \cdot \log(\frac{1}{\varepsilon\,\gamma\,\delta})$, there exists a learning problem (with a set of linear regression functions) satisfying Definition 2.15/Definition 2.16 such that (1) any "uncertainty-based" active learner suffers*

expected standard excess error $\Omega(B^{-1})$*; yet* (2) *with probability at least* $1-\delta$*, Algorithm 1 returns a classifier with standard excess error at most* ε*.*

The above result demonstrates the superiority of our Algorithm 1 over any "uncertainty-based" active learner. Moreover, we show that Algorithm 1 achieves similar guarantees as in Theorem 2.13 under the strictly harder noise-seeking conditions. Specifically, we have the following guarantees.

Theorem 2.18. *With an appropriate choice of the abstention parameter* γ *in Algorithm 1 and randomization over the abstention region, Algorithm 1 learns a classifier* $\check{h}$ *with* $\varepsilon + \zeta_0$ *standard excess error after querying* $\widetilde{\Theta}(\tau_0^{-2})$ *labels under Definition 2.15 or querying* $\widetilde{\Theta}(\varepsilon^{-2/(1+\beta)})$ *labels under Definition 2.16.*

The special case of the noise-seeking condition with $\zeta_0 = 0$ is recently studied in (Kpotufe et al., 2021), where the authors conclude that no active learners can outperform the passive counterparts in the *nonparametric* regime. Theorem 2.18 shows that, in the *parametric* setting (with function approximation), Algorithm 1 provably overcomes these noise-seeking conditions.

2.4 Extensions

We provide two adaptations of our main algorithm (Algorithm 1) that can (1) achieve constant label complexity for a general set of regression functions (Section 2.4.1); and (2) adapt to model misspecification (Section 2.4.2). These two adaptations can also be efficiently implemented via regression oracle and enjoy similar guarantees stated in Theorem 2.8. We defer computational analysis to Section 2.5.5 and Section 2.5.6.

2.4.1 Constant Label Complexity

We start by considering a simple problem instance with $\mathcal{X} = \{x\}$, where active learning is reduced to mean estimation of $\eta(x)$. Consider the Massart noise case where $\eta(x) \notin [\frac{1}{2} - \tau_0, \frac{1}{2} + \tau_0]$. No matter how small the desired accuracy level $\varepsilon > 0$

is, the learner should not spend more than $O(\frac{\log(1/\delta)}{\tau_0^2})$ labels to correctly classify x with probability at least $1-\delta$, which ensures 0 excess error. In the general setting, but with Chow's excess error, a similar result follows: It takes at most $O(\frac{\log(1/\delta)}{\gamma^2})$ samples to verify if $\eta(x)$ is contained in $[\frac{1}{2}-\gamma, \frac{1}{2}+\gamma]$ or not. Taking the optimal action within $\{+1, -1, \perp\}$ (based on Eq. (2.2)) then leads to 0 Chow's excess error. This reasoning shows that, at least in simple cases, one should be able to achieve *constant* label complexity no matter how small ε is. One natural question to ask is as follows.

Can active learning achieve constant label complexity in more general cases?

We provide the first affirmative answer to the above question with a *general* set of regression function $\mathcal{F}$ (finite), and under *general* action space $\mathcal{X}$ and marginal distribution $\mathcal{D}_{\mathcal{X}}$. The positive result is achieved by Algorithm 2 (deferred to Section 2.5.5.2), which differs from Algorithm 1 in two aspects: (1) we drop the epoch scheduling, and (2) apply a tighter elimination step derived from an optimal stopping theorem. Another change comes from the analysis of the algorithm: Instead of analyzing with respect to the disagreement coefficient, we work with the *eluder dimension* $\mathfrak{e} := \sup_{f^\star \in \mathcal{F}} \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2)$.[7] To do that, we analyze active learning from the perspective of *regret minimization with selective querying* (Dekel et al., 2012), which allows us to incorporate techniques developed in the field of contextual bandits (Russo and Van Roy, 2013; Foster et al., 2020c). We defer a detailed discussion to Section 2.5.5.1 and provide the following guarantees.

Theorem 2.19. *With probability at least* $1-2\delta$, *Algorithm 2 returns a classifier with expected Chow's excess error at most* ε *and label complexity* $O(\frac{\mathfrak{e}\cdot\log(|\mathcal{F}|/\delta)}{\gamma^2})$, *which is independent of* $\frac{1}{\varepsilon}$.

Based on discussion in Section 2.3, we can immediately translate the above results into *standard* excess error guarantees under the Massart noise (with γ re-

[7] We formally define eluder dimension in Section 2.5.1. As examples, we have $\mathfrak{e} = O(d \cdot \log \frac{1}{\gamma})$ for linear functions in $\mathbb{R}^d$, and $\mathfrak{e} = O(C_{\mathsf{link}} \cdot d \log \frac{1}{\gamma})$ for generalized linear functions (where C_{link} is a quantity related to the link function).

placed by τ_0). We next discuss why existing algorithms/analyses do not guarantee constant label complexity, even in the linear case.

1. **Epoch scheduling.** Many algorithms proceed in epochs and aim at *halving* the excess error after each epoch (Balcan et al., 2007; Zhang and Chaudhuri, 2014; Puchkin and Zhivotovskiy, 2021). One inevitably needs $\log \frac{1}{\varepsilon}$ epochs to achieve ε excess error.

2. **Relating to disagreement coefficient.** The algorithm presented in Krishnamurthy et al. (2019) does not use epoch scheduling. However, their label complexity are analyzed with disagreement coefficient, which incurs an $\sum_{t=1}^{1/\varepsilon} \frac{1}{t} = O(\log \frac{1}{\varepsilon})$ term in the label complexity.

Remark 2.20. *Algorithm 2 also provides guarantees when x is selected by an adaptive adversary (instead of i.i.d. sampled $x \sim \mathcal{D}_X$). In that case, we simultaneously upper bound the regret and the label complexity (see Theorem 2.42 in Section 2.5.5.2). Our results can be viewed as a generalization of the results developed in the linear case (Dekel et al., 2012).*

2.4.2 Dealing with Model Misspecification

Our main results are developed under realizability (Assumption 2.1), which assumes that there exists a $f^\star \in \mathcal{F}$ such that $f^\star = \eta$. In this section, we relax that assumption and allow model misspecification. We assume the learner is given a set of regression function $\mathcal{F} : \mathcal{X} \to [0, 1]$ that may only *approximates* the conditional probability η. More specifically, we make the following assumption.

Assumption 2.21 (Model misspecification)**.** *There exists a $\bar{f} \in \mathcal{F}$ such that $\bar{f}$ approximate η up to $\kappa > 0$ accuracy, i.e., $\sup_{x \in \mathcal{X}} \left|\bar{f}(x) - \eta(x)\right| \leqslant \kappa$.*

We use a variation of Algorithm 1 to adapt to model misspecification (Algorithm 3, deferred to Section 2.5.6.1). Compared to Algorithm 1, the main change in Algorithm 3 is to apply a more conservative step in determining the active set $\mathcal{F}_m$ at each epoch: We maintain a larger active set of regression function to ensure that $\bar{f}$ is not eliminated throughout all epochs. Our algorithm proceeds *without* knowing

the misspecification level κ. However, the excess error bound presented next holds under the condition that $\kappa \leqslant \varepsilon$ (i.e., it requires that the misspecification is no larger than the desired accuracy; we leave a comprehensive study for the problem with *general* approximation error for future work). Abbreviate $\bar{\theta} := \sup_{\iota>0} \theta^{\mathrm{val}}_{\bar{f}}(\mathcal{F}, \gamma/2, \iota)$, we achieve the following guarantees.

Theorem 2.22. *Suppose $\kappa \leqslant \varepsilon$. With probability at least $1-2\delta$, Algorithm 3 returns a classifier with Chow's excess error $O(\varepsilon \cdot \bar{\theta} \cdot \log(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}))$ and label complexity $O(\frac{\bar{\theta}\,\mathrm{Pdim}(\mathcal{F})}{\gamma^2} \cdot \log^2(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}) \cdot \log(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}))$.*

2.5 Proofs and Supporting Results

2.5.1 Disagreement Coefficient, Star Number and Eluder Dimension

We provide formal definitions/guarantees of value function disagreement coefficient, eluder dimension and star number in this section. These results are developed in Foster et al. (2020c); Russo and Van Roy (2013). Since our guarantees are developed in terms of these complexity measures, any future developments on these complexity measures (e.g., with respect to richer function classes) directly lead to broader applications of our algorithms.

We first state known upper bound on value function disagreement coefficient with respect to nice sets of regression functions.

Proposition 2.23 (Foster et al. (2020c))**.** *For any $f^\star \in \mathcal{F}$ and $\gamma, \varepsilon > 0$, let $\theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma, \varepsilon)$ be the value function disagreement coefficient defined in Definition 2.6. Let $\phi : \mathcal{X} \to \mathbb{R}^d$ be a fixed feature mapping and $\mathcal{W} \subseteq \mathbb{R}^d$ be a fixed set. The following upper bounds hold true.*

- *Suppose $\mathcal{F} := \{x \mapsto \langle \phi(x), w\rangle : w \in \mathcal{W}\}$ is a set of linear functions. We then have $\sup_{f\in\mathcal{F},\gamma>0,\varepsilon>0} \theta^{\mathrm{val}}_f(\mathcal{F}, \gamma, \varepsilon) \leqslant d$.*

- *Suppose* $\mathcal{F} := \{x \mapsto \sigma(\langle \phi(x), w\rangle) : w \in \mathcal{W}\}$ *is a set of generalized linear functions with any fixed link function* $\sigma : \mathbb{R} \to \mathbb{R}$ *such that* $0 < c_l < \sigma' \leqslant c_u$. *We then have* $\sup_{f\in\mathcal{F},\gamma>0,\varepsilon>0} \theta_f^{\mathrm{val}}(\mathcal{F},\gamma,\varepsilon) \leqslant \frac{c_u}{c_l}^2 \cdot d$.

We next provide the formal definition of value function eluder dimension and star number (Foster et al., 2020c; Russo and Van Roy, 2013).

Definition 2.24 (Value function eluder dimension)**.** *For any* $f^\star \in \mathcal{F}$ *and* $\gamma > 0$, *let* $\check{\mathfrak{e}}_{f^\star}(\mathcal{F},\gamma)$ *be the length of the longest sequence of data points* $x^1,\dots,x^m$ *such that for all* i, *there exists* $f^i \in \mathcal{F}$ *such that*

$$|f^i(x^i) - f^\star(x^i)| > \gamma, \quad \textit{and} \quad \sum_{j<i} (f^i(x^j) - f^\star(x^j))^2 \leqslant \gamma^2.$$

The value function eluder dimension is defined as $\mathfrak{e}_{f^\star}(\mathcal{F},\gamma_0) := \sup_{\gamma\geqslant\gamma_0} \check{\mathfrak{e}}_{f^\star}(\mathcal{F},\gamma)$.

Definition 2.25 (Value function star number)**.** *For any* $f^\star \in \mathcal{F}$ *and* $\gamma > 0$, *let* $\check{\mathfrak{s}}_{f^\star}(\mathcal{F},\gamma)$ *be the length of the longest sequence of data points* $x^1,\dots,x^m$ *such that for all* i, *there exists* $f^i \in \mathcal{F}$ *such that*

$$|f^i(x^i) - f^\star(x^i)| > \gamma, \quad \textit{and} \quad \sum_{j\neq i} (f^i(x^j) - f^\star(x^j))^2 \leqslant \gamma^2.$$

The value function eluder dimension is defined as $\mathfrak{s}_{f^\star}(\mathcal{F},\gamma_0) := \sup_{\gamma\geqslant\gamma_0} \check{\mathfrak{s}}_{f^\star}(\mathcal{F},\gamma)$.

Since the second constrain in the definition of star number is more stringent than the counterpart in the definition of eluder dimension, one immediately have that $\mathfrak{s}_{f^\star}(\mathcal{F},\gamma) \leqslant \mathfrak{e}_{f^\star}(\mathcal{F},\gamma)$. We provide known upper bounds for eluder dimension next.

Proposition 2.26 (Russo and Van Roy (2013))**.** *Let* $\phi : \mathcal{X} \to \mathbb{R}^d$ *be a fixed feature mapping and* $\mathcal{W} \subseteq \mathbb{R}^d$ *be a fixed set. Suppose* $\sup_{x\in\mathcal{X}} \|\phi(x)\|_2 \leqslant 1$ *and* $\sup_{w\in\mathcal{W}} \|w\|_2 \leqslant 1$. *The following upper bounds hold true.*

- *Suppose* $\mathcal{F} := \{x \mapsto \langle \phi(x), w\rangle : w \in \mathcal{W}\}$ *is a set of linear functions. We then have* $\sup_{f^\star\in\mathcal{F}} \mathfrak{e}_{f^\star}(\mathcal{F},\gamma) = O(d \log \frac{1}{\gamma})$.

- *Suppose* $\mathcal{F} := \{x \mapsto \sigma(\langle \phi(x), w\rangle) : w \in \mathcal{W}\}$ *is a set of generalized linear functions with any fixed link function* $\sigma : \mathbb{R} \to \mathbb{R}$ *such that* $0 < c_l < \sigma' \leqslant c_u$. *We then have* $\sup_{f^\star \in \mathcal{F}} \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma) = O\big(\big(\frac{c_u}{c_l}\big)^2 d \log\big(\frac{c_u}{\gamma}\big)\big)$.

The next result shows that the disagreement coefficient (with our Definition 2.6) can be always upper bounded by (squared) star number and eluder dimension.

Proposition 2.27 (Foster et al. (2020c))**.** *Suppose* $\mathcal{F}$ *is a uniform Glivenko-Cantelli class.*

For any $f^\star : \mathcal{X} \to [0,1]$ *and* $\gamma, \varepsilon > 0$, *we have* $\theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma, \varepsilon) \leqslant 4(\mathfrak{s}_{f^\star}(\mathcal{F}, \gamma))^2$, *and* $\theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma, \varepsilon) \leqslant 4\, \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma)$.

The requirement that $\mathcal{F}$ is a uniform Glivenko-Cantelli class is rather weak: It is satisfied as long as $\mathcal{F}$ has finite Pseudo dimension (Anthony, 2002).

In our analysis, we sometimes work with sub probability measure (due to selective sampling). Our next result shows that defining the disagreement coefficient over all (sub) probability measures will not affect its value. More specifically, denote $\widetilde{\theta}^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma, \varepsilon)$ be the disagreement coefficient defined in Definition 2.6, but with sup taking over all probability and sub probability measures. We then have the following equivalence.

Proposition 2.28. *Fix any* $\gamma_0, \varepsilon_0 \geqslant 0$. *We have* $\widetilde{\theta}^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0) = \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0)$.

Proof. We clearly have $\widetilde{\theta}^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0) \geqslant \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0)$ by additionally considering sub probability measures. We next show the opposite direction.

Fix any sub probability measure $\widetilde{\mathcal{D}}_X$ that is non-zero (otherwise we have $\mathbb{P}_{x \sim \widetilde{\mathcal{D}}_X}(\cdot) = 0$). Suppose $\mathbb{E}_{x \sim \widetilde{\mathcal{D}}_X}[1] = \kappa < 1$. We can now consider its normalized probability measure $\overline{\mathcal{D}}_X$ such that $\overline{\mathcal{D}}_X(\omega) = \frac{\widetilde{\mathcal{D}}_X(\omega)}{\kappa}$ (for any ω in the sigma algebra). Now fix any $\gamma > \gamma_0$ and $\varepsilon > \varepsilon_0$. We have

$$\begin{aligned}
&\frac{\gamma^2}{\varepsilon^2} \cdot \mathbb{P}_{\widetilde{\mathcal{D}}_X}\Big(\exists f \in \mathcal{F} : |f(x) - f^\star(x)| > \gamma, \|f - f^\star\|^2_{\widetilde{\mathcal{D}}_X} \leqslant \varepsilon^2\Big) \\
&= \frac{\gamma^2}{\varepsilon^2/\kappa} \cdot \mathbb{P}_{\overline{\mathcal{D}}_X}\Big(\exists f \in \mathcal{F} : |f(x) - f^\star(x)| > \gamma, \|f - f^\star\|^2_{\overline{\mathcal{D}}_X} \leqslant \varepsilon^2/\kappa\Big)
\end{aligned}$$

$$= \frac{\gamma^2}{\bar{\varepsilon}^2} \cdot \mathbb{P}_{\overline{\mathcal{D}}_X}\left(\exists f \in \mathcal{F} : |f(x) - f^\star(x)| > \gamma, \|f - f^\star\|^2_{\overline{\mathcal{D}}_X} \leqslant \bar{\varepsilon}^2\right)$$
$$\leqslant \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0),$$

where we denote $\bar{\varepsilon} := \frac{\varepsilon}{\sqrt{\kappa}} > \varepsilon$, and the last follows from the fact that $\overline{\mathcal{D}}_X$ is a probability measure. We then have $\widetilde{\theta}^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0) \leqslant \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0)$, and thus the desired result. □

2.5.2 Concentration Results

The Freedman's inequality is quite common in the field of active learning and contextual bandits, e.g., (Freedman, 1975; Agarwal et al., 2014; Krishnamurthy et al., 2019; Foster et al., 2020c). We thus state the result without proof.

Lemma 2.29 (Freedman's inequality)**.** *Let $(Z_t)_{t\leqslant T}$ be a real-valued martingale difference sequence adapted to a filtration $\mathfrak{F}_t$, and let $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot \mid \mathfrak{F}_{t-1}]$. If $|Z_t| \leqslant B$ almost surely, then for any $\eta \in (0, 1/B)$ it holds with probability at least $1-\delta$,*

$$\sum_{t=1}^{T} Z_t \leqslant \eta \sum_{t=1}^{T} \mathbb{E}_t[Z_t^2] + \frac{\log \delta^{-1}}{\eta}.$$

Lemma 2.30. *Let $(Z_t)_{t\leqslant T}$ to be real-valued sequence of random variables adapted to a filtration $\mathfrak{F}_t$. If $|Z_t| \leqslant B$ almost surely, then with probability at least $1-\delta$,*

$$\sum_{t=1}^{T} Z_t \leqslant \frac{3}{2} \sum_{t=1}^{T} \mathbb{E}_t[Z_t] + 4B \log(2\delta^{-1}),$$

and

$$\sum_{t=1}^{T} \mathbb{E}_t[Z_t] \leqslant 2 \sum_{t=1}^{T} Z_t + 8B \log(2\delta^{-1}).$$

Proof. This is a direct consequence of Lemma 2.29. □

We define/recall some notations first. Fix any epoch $m \in [M]$ and any time step t within epoch m. For any $f \in \mathcal{F}$, we denote $M_t(f) := Q_t((f(x_t)-y_t)^2-(f^\star(x_t)-y_t)^2)$, and $\widehat{R}_m(f) := \sum_{t=1}^{\tau_{m-1}} Q_t(f(x_t)-y_t)^2$. Recall that we have $Q_t = g_m(x_t)$. We define filtration $\mathfrak{F}_t := \sigma((x_1,y_1),\dots,(x_t,y_t))$,[8] and denote $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot \mid \mathfrak{F}_{t-1}]$.

We first provide a simple concentration result with respect to a finite $\mathcal{F}$.

Lemma 2.31. *Suppose $\mathcal{F}$ is finite. Fix any $\delta \in (0,1)$. For any $\tau, \tau' \in [T]$ such that $\tau < \tau'$, with probability at least $1-\delta$, we have*

$$\sum_{t=\tau}^{\tau'} M_t(f) \leqslant \sum_{t=\tau}^{\tau'} \frac{3}{2}\mathbb{E}_t[M_t(f)] + C_\delta(\mathcal{F}),$$

and

$$\sum_{t=\tau}^{\tau'} \mathbb{E}_t[M_t(f)] \leqslant 2\sum_{t=\tau}^{\tau'} M_t(f) + C_\delta(\mathcal{F}),$$

where $C_\delta(\mathcal{F}) = 8\log\left(\frac{|\mathcal{F}|\cdot T^2}{\delta}\right)$.

Proof. We first notice that $M_t(f)$ adapts to filtration $\mathfrak{F}_t$, and satisfies $|M_t(f)| \leqslant 1$. The results follow by taking Lemma 2.30 together with a union bound over $f \in \mathcal{F}$ and $\tau, \tau' \in [T]$. □

Although one can not directly apply a union bound as in Lemma 2.31 in the case when the set of regression function $\mathcal{F}$ is infinite (but has finite Pseudo dimension by assumption), it turns out that similar guarantees as in Lemma 2.31 can be derived. We first recall the formal definition of the Pseudo dimension of $\mathcal{F}$.

Definition 2.32 (Pseudo Dimension, Pollard (1984); Haussler (1989, 1995))**.** *Consider a set of real-valued function $\mathcal{F} : \mathcal{X} \to \mathbb{R}$. The pseudo-dimension* $\mathrm{Pdim}(\mathcal{F})$ *of $\mathcal{F}$ is defined as the VC dimension of the set of threshold functions $\{(x,\zeta) \mapsto \mathbb{1}(f(x) > \zeta) : f \in \mathcal{F}\}$.*

[8] y_t is not observed (and thus not included in the filtration) when $Q_t = 0$. Note that Q_t is measurable with respect to $\sigma((\mathfrak{F}_{t-1}, x_t))$.

Lemma 2.33 (Krishnamurthy et al. (2019)). *Suppose* $\mathrm{Pdim}(\mathcal{F}) < \infty$. *Fix any* $\delta \in (0,1)$. *For any* $\tau, \tau' \in [T]$ *such that* $\tau < \tau'$, *with probability at least* $1-\delta$, *we have*

$$\sum_{t=\tau}^{\tau'} M_t(f) \leqslant \sum_{t=\tau}^{\tau'} \frac{3}{2}\mathbb{E}_t[M_t(f)] + C_\delta(\mathcal{F}),$$

and

$$\sum_{t=\tau}^{\tau'} \mathbb{E}_t[M_t(f)] \leqslant 2\sum_{t=\tau}^{\tau'} M_t(f) + C_\delta(\mathcal{F}),$$

where $C_\delta(\mathcal{F}) = C \cdot \left(\mathrm{Pdim}(\mathcal{F}) \cdot \log T + \log\left(\frac{\mathrm{Pdim}(\mathcal{F})\cdot T}{\delta}\right)\right) \leqslant C' \cdot \left(\mathrm{Pdim}(\mathcal{F}) \cdot \log\left(\frac{T}{\delta}\right)\right)$, *where* $C, C' > 0$ *are universal constants.*

We will be primarily using Lemma 2.33 in the following. However, one can replace Lemma 2.33 with Lemma 2.31 to derive results with respect to a finite set of regressions $\mathcal{F}$.

2.5.3 Proofs and Supporting Results for Section 2.2

We give the proof of Theorem 2.7 and Theorem 2.8. Supporting lemmas used in the proofs are deferred to Section 2.5.3.1.

Fix any classifier $\widehat{h} : \mathcal{X} \to \{+1, -1, \perp\}$. For any $x \in \mathcal{X}$, we introduce the notion

$$\begin{aligned}
\mathsf{excess}_\gamma(\widehat{h}; x) &:= \mathbb{P}_{y|x}\big(y \neq \mathrm{sign}(\widehat{h}(x))\big) \cdot \mathbb{1}\big(\widehat{h}(x) \neq \perp\big) + \big(1/2 - \gamma\big) \cdot \mathbb{1}\big(\widehat{h}(x) = \perp\big) \\
&\quad - \mathbb{P}_{y|x}\big(y \neq \mathrm{sign}(h^\star(x))\big) \\
&= \mathbb{1}\big(\widehat{h}(x) \neq \perp\big) \cdot \big(\mathbb{P}_{y|x}\big(y \neq \mathrm{sign}(\widehat{h}(x))\big) - \mathbb{P}_{y|x}\big(y \neq \mathrm{sign}(h^\star(x))\big)\big) \\
&\quad + \mathbb{1}\big(\widehat{h}(x) = \perp\big) \cdot \big(\big(1/2 - \gamma\big) - \mathbb{P}_{y|x}\big(y \neq \mathrm{sign}(h^\star(x))\big)\big) \qquad (2.6)
\end{aligned}$$

to represent the excess error of $\widehat{h}$ at point $x \in \mathcal{X}$. Excess error of classifier $\widehat{h}$ can be then written as $\mathsf{excess}_\gamma(\widehat{h}) := \mathrm{err}_\gamma(\widehat{h}) - \mathrm{err}(h^\star) = \mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathsf{excess}_\gamma(\widehat{h}; x)]$.

Theorem 2.7. *With probability at least* $1-2\delta$, *Algorithm 1 returns a classifier with Chow's excess error at most* ε *and label complexity* $O(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\gamma^2} \cdot \log^2(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma}) \cdot \log(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma \delta}))$.

Proof. We analyze under the good event $\mathcal{E}$ defined in Lemma 2.33, which holds with probability at least $1-\delta$. Note that all supporting lemmas stated in Section 2.5.3.1 hold true under this event.

We analyze the Chow's excess error of $\widehat{h}_m$, which is measurable with respect to $\mathfrak{F}_{\tau_{m-1}}$. For any $x \in \mathcal{X}$, if $g_m(x) = 0$, Lemma 2.39 implies that $\operatorname{excess}_\gamma(\widehat{h}_m; x) \leqslant 0$. If $g_m(x) = 1$, we know that $\widehat{h}_m(x) \neq \perp$ and $\frac{1}{2} \in (\operatorname{lcb}(x; \mathcal{F}_m), \operatorname{ucb}(x; \mathcal{F}_m))$. Note that $\widehat{h}_m(x) \neq h^\star(x)$ only if $\operatorname{sign}(2f^\star(x)-1) \cdot \operatorname{sign}(2\widehat{f}_m(x)-1) \leqslant 0$. Since $f^\star, \widehat{f}_m \in \mathcal{F}_m$ by Lemma 2.35. The error incurred in this case can be upper bounded by $2|f^\star(x) - 1/2| \leqslant 2w(x; \mathcal{F}_m)$, which results in $\operatorname{excess}_\gamma(\widehat{h}_m; x) \leqslant 2w(x; \mathcal{F}_m)$. Combining these two cases together, we have

$$\operatorname{excess}_\gamma(\widehat{h}_m) \leqslant 2\mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) = 1) \cdot w(x; \mathcal{F}_m)].$$

Take $m = M$ and apply Lemma 2.38, with notation $\rho_m := 2\beta_m + C_\delta$, leads to the following guarantee.

$$\begin{aligned}\operatorname{excess}_\gamma(\widehat{h}_M) &\leqslant \frac{8\rho_M}{\tau_{M-1}\gamma} \cdot \theta^{\mathrm{val}}_{f^\star}\left(\mathcal{F}, \gamma/2, \sqrt{\rho_M/2\tau_{M-1}}\right) \\ &= O\left(\frac{\operatorname{Pdim}(\mathcal{F}) \cdot \log(T/\delta)}{T\gamma} \cdot \theta^{\mathrm{val}}_{f^\star}\left(\mathcal{F}, \gamma/2, \sqrt{C_\delta/T}\right)\right),\end{aligned}$$

where we use the fact that $\frac{T}{2} \leqslant \tau_{M-1} \leqslant T$ and definitions of β_m and C_δ. Simply considering $\theta := \sup_{f^\star\in\mathcal{F}, \iota>0} \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma/2, \iota)$ as an upper bound of $\theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma/2, \sqrt{C_\delta/T})$ and taking

$$T = O\left(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon\gamma} \cdot \log\left(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon\gamma\delta}\right)\right)$$

ensures that $\operatorname{excess}_\gamma(\widehat{h}_M) \leqslant \varepsilon$.

We now analyze the label complexity (note that the sampling process of Algorithm 1 stops at time $t = \tau_{M-1}$). Note that $\mathbb{E}[\mathbb{1}(Q_t = 1) \mid \mathfrak{F}_{t-1}] = \mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) =$

1)] for any epoch $m \geqslant 2$ and time step t within epoch m. Combine Lemma 2.30 with Lemma 2.37 leads to

$$
\begin{aligned}
&\sum_{t=1}^{\tau_{M-1}} \mathbb{1}(Q_t = 1) \\
&\leqslant \frac{3}{2} \sum_{t=1}^{\tau_{M-1}} \mathbb{E}[\mathbb{1}(Q_t = 1) \mid \mathfrak{F}_{t-1}] + 4 \log \delta^{-1} \\
&\leqslant 3 + \frac{3}{2} \sum_{m=2}^{M-1} \frac{(\tau_m - \tau_{m-1}) \cdot 4\rho_m}{\tau_{m-1}\gamma^2} \cdot \theta^{\mathrm{val}}_{f^\star}\left(\mathcal{F}, \gamma/2, \sqrt{\rho_m/2\tau_{m-1}}\right) + 4 \log \delta^{-1} \\
&\leqslant 3 + 6 \sum_{m=2}^{M-1} \frac{\rho_m}{\gamma^2} \cdot \theta^{\mathrm{val}}_{f^\star}\left(\mathcal{F}, \gamma/2, \sqrt{\rho_m/2\tau_{m-1}}\right) + 4 \log \delta^{-1} \\
&\leqslant 3 + 4 \log \delta^{-1} + \frac{18 \log T \cdot M \cdot C_\delta}{\gamma^2} \cdot \theta^{\mathrm{val}}_{f^\star}\left(\mathcal{F}, \gamma/2, \sqrt{C_\delta/T}\right) \\
&= O\left(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\gamma^2} \cdot \left(\log\left(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma}\right)\right)^2 \cdot \log\left(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma \delta}\right)\right),
\end{aligned}
$$

with probability at least $1 - 2\delta$ (due to an additional application of Lemma 2.30); where we plug the above choice of T and upper bound other terms as before. □

A slightly different guarantee for Algorithm 1. The stated Algorithm 1 takes $\theta := \sup_{f^\star \in \mathcal{F}, \iota > 0} \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma/2, \iota)$ as an input (the value of θ can be upper bounded for many function class $\mathcal{F}$, as discussed in Section 2.5.1). However, we don't necessarily need to take θ as an input to the algorithm. Indeed, we can simply run a modified version of Algorithm 1 with $T = \frac{\operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma}$. Following similar analyses in proof of Theorem 2.7, set $\iota := \sqrt{C_\delta/T} \propto \sqrt{\gamma\varepsilon}$, the modified version achieves excess error

$$
\operatorname{excess}_\gamma(\widehat{h}_M) = O\left(\varepsilon \cdot \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma/2, \iota) \cdot \log\left(\frac{\operatorname{Pdim}(\mathcal{F})}{\varepsilon \delta \gamma}\right)\right)
$$

with label complexity

$$O\left(\frac{\theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma/2, \iota) \cdot \mathrm{Pdim}(\mathcal{F})}{\gamma^2} \cdot \left(\log\left(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}\right)\right)^2 \cdot \log\left(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}\right)\right).$$

We now discuss the efficient implementation of Algorithm 1 and its computational complexity. We first state some known results in computing the confidence intervals with respect to a set of regression functions $\mathcal{F}$.

Proposition 2.34 (Krishnamurthy et al. (2017); Foster et al. (2018, 2020c))**.** *Consider the setting studied in Algorithm 1. Fix any epoch $m \in [M]$ and denote $\mathcal{B}_m := \{(x_t, Q_t, y_t)\}_{t=1}^{\tau_{m-1}}$. Fix any $\alpha > 0$. For any data point $x \in \mathcal{X}$, there exists algorithms $\mathbf{Alg}_{\mathsf{lcb}}$ and $\mathbf{Alg}_{\mathsf{ucb}}$ that certify*

$$\mathsf{lcb}(x; \mathcal{F}_m) - \alpha \leqslant \mathbf{Alg}_{\mathsf{lcb}}(x; \mathcal{B}_m, \beta_m, \alpha) \leqslant \mathsf{lcb}(x; \mathcal{F}_m) \quad \textit{and}$$
$$\mathsf{ucb}(x; \mathcal{F}_m) \leqslant \mathbf{Alg}_{\mathsf{ucb}}(x; \mathcal{B}_m, \beta_m, \alpha) \leqslant \mathsf{ucb}(x; \mathcal{F}_m) + \alpha.$$

The algorithms take $O(\frac{1}{\alpha^2} \log \frac{1}{\alpha})$ calls of the regression oracle for general $\mathcal{F}$ and take $O(\log \frac{1}{\alpha})$ calls of the regression oracle if $\mathcal{F}$ is convex and closed under pointwise convergence.

Proof. See Algorithm 2 in Krishnamurthy et al. (2017) for the general case; and Algorithm 3 in Foster et al. (2018) for the case when $\mathcal{F}$ is convex and closed under pointwise convergence. □

We next discuss the computational efficiency of Algorithm 1. Recall that we redefine $\theta := \sup_{f^\star \in \mathcal{F}, \iota > 0} \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma/4, \iota)$ in the Theorem 2.8 to account to approximation error.

Theorem 2.8. *Algorithm 1 can be efficiently implemented via the regression oracle and enjoys the same theoretical guarantees stated in Theorem 2.7. The number of oracle calls needed is $\widetilde{O}(\frac{\theta\,\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma^3})$ for a general set of regression functions $\mathcal{F}$, and $\widetilde{O}(\frac{\theta\,\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma})$ when $\mathcal{F}$ is convex and closed under pointwise convergence. The per-example inference time of the*

learned $\widehat{h}_M$ *is* $\widetilde{O}(\frac{1}{\gamma^2}\log^2(\frac{\theta\,\mathrm{Pdim}(\mathcal{F})}{\varepsilon}))$ *for general* $\mathcal{F}$*, and* $\widetilde{O}(\log\frac{1}{\gamma})$ *when* $\mathcal{F}$ *is convex and closed under pointwise convergence.*

Proof. Fix any epoch $m \in [M]$. Denote $\bar{\alpha} := \frac{\gamma}{4M}$ and $\alpha_m := \frac{(M-m)\gamma}{4M}$. With any observed $x \in \mathcal{X}$, we construct the approximated confidence intervals $\widehat{\mathsf{lcb}}(x;\mathcal{F}_m)$ and $\widehat{\mathsf{ucb}}(x;\mathcal{F}_m)$ as follows.

$$\widehat{\mathsf{lcb}}(x;\mathcal{F}_m) := \mathbf{Alg}_{\mathsf{lcb}}(x;\mathcal{B}_m,\beta_m,\bar{\alpha}) - \alpha_m \quad \text{and}$$
$$\widehat{\mathsf{ucb}}(x;\mathcal{F}_m) := \mathbf{Alg}_{\mathsf{ucb}}(x;\mathcal{B}_m,\beta_m,\bar{\alpha}) + \alpha_m.$$

For efficient implementation of Algorithm 1, we replace $\mathsf{lcb}(x;\mathcal{F}_m)$ and $\mathsf{ucb}(x;\mathcal{F}_m)$ with $\widehat{\mathsf{lcb}}(x;\mathcal{F}_m)$ and $\widehat{\mathsf{ucb}}(x;\mathcal{F}_m)$ in the construction of $\widehat{h}_m$ and g_m.

Based on Proposition 2.34, we know that

$$\mathsf{lcb}(x;\mathcal{F}_m) - \alpha_m - \bar{\alpha} \leqslant \widehat{\mathsf{lcb}}(x;\mathcal{F}_m) \leqslant \mathsf{lcb}(x;\mathcal{F}_m) - \alpha_m \quad \text{and}$$
$$\mathsf{ucb}(x;\mathcal{F}_m) + \alpha_m \leqslant \widehat{\mathsf{ucb}}(x;\mathcal{F}_m) \leqslant \mathsf{ucb}(x;\mathcal{F}_m) + \alpha_m + \bar{\alpha}.$$

Since $\alpha_m + \bar{\alpha} \leqslant \frac{\gamma}{4}$ for any $m \in [M]$, the guarantee in Lemma 2.36 can be modified as $g_m(x) = 1 \implies w(x;\mathcal{F}_m) \geqslant \frac{\gamma}{2}$.

Fix any $m \geqslant 2$. Since $\mathcal{F}_m \subseteq \mathcal{F}_{m-1}$ by Lemma 2.35, we have

$$\widehat{\mathsf{lcb}}(x;\mathcal{F}_m) \geqslant \mathsf{lcb}(x;\mathcal{F}_m) - \alpha_m - \bar{\alpha} \geqslant \mathsf{lcb}(x;\mathcal{F}_{m-1}) - \alpha_{m-1} \geqslant \widehat{\mathsf{lcb}}(x;\mathcal{F}_{m-1}) \quad \text{and}$$
$$\widehat{\mathsf{ucb}}(x;\mathcal{F}_m) \leqslant \mathsf{ucb}(x;\mathcal{F}_m) + \alpha_m + \bar{\alpha} \leqslant \mathsf{ucb}(x;\mathcal{F}_{m-1}) + \alpha_{m-1} \leqslant \widehat{\mathsf{ucb}}(x;\mathcal{F}_{m-1}).$$

These ensure $\mathbb{1}(g_m(x) = 1) \leqslant \mathbb{1}(g_{m-1}(x) = 1)$. Thus, the guarantees stated in Lemma 2.37 and Lemma 2.38 still hold (with $\frac{\gamma}{2}$ replaced by $\frac{\gamma}{4}$ due to modification of Lemma 2.36). The guarantee stated in Lemma 2.39 also hold since $\widehat{\mathsf{lcb}}(x;\mathcal{F}_m) \leqslant \mathsf{lcb}(x;\mathcal{F}_m)$ and $\widehat{\mathsf{ucb}}(x;\mathcal{F}_m) \geqslant \mathsf{ucb}(x;\mathcal{F}_m)$ by construction. As a result, the guarantees stated in Theorem 2.7 hold true with changes only in constant terms.

We now discuss the computational complexity of the efficient implementation. At the beginning of each epoch m. We use one oracle call to compute $\widehat{f}_m = \arg\min_{f\in\mathcal{F}} \sum_{t=1}^{\tau_{m-1}} Q_t(f(x_t) - y_t)^2$. The main computational cost comes from

computing $\widehat{\text{lcb}}$ and $\widehat{\text{ucb}}$ at each time step. We take $\alpha = \overline{\alpha} := \frac{\gamma}{4M}$ into Proposition 2.34, which leads to $O(\frac{(\log T)^2}{\gamma^2} \cdot \log(\frac{\log T}{\gamma}))$ calls of the regression oracle for general $\mathcal{F}$ and $O(\log(\frac{\log T}{\gamma}))$ calls of the regression oracle for any convex $\mathcal{F}$ that is closed under pointwise convergence. This also serves as the per-example inference time for $\widehat{h}_M$. The total computational cost of Algorithm 1 is then derived by multiplying the per-round cost by T and plugging $T = \widetilde{O}(\frac{\theta \operatorname{Pdim}(\mathcal{F})}{\varepsilon \gamma})$ into the bound (for any parameter, we only keep poly factors in the total computational cost and keep poly or polylog dependence in the per-example computational cost). □

2.5.3.1 Supporting Lemmas

We use $\mathcal{E}$ to denote the good event considered in Lemma 2.33, and analyze under this event in this section. We abbreviate $C_\delta := C_\delta(\mathcal{F})$ in the following analysis.

Lemma 2.35. *The followings hold true:*

1. $f^\star \in \mathcal{F}_m$ *for any* $m \in [M]$.
2. $\sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t[M_t(f)] \leqslant 2\beta_m + C_\delta$ *for any* $f \in \mathcal{F}_m$.
3. $\mathcal{F}_{m+1} \subseteq \mathcal{F}_m$ *for any* $m \in [M-1]$.

Proof. 1. Fix any epoch $m \in [M]$ and time step t within epoch m. Since $\mathbb{E}[y_t] = f^\star(x_t)$, we have $\mathbb{E}_t[M_t(f)] = \mathbb{E}[Q_t(f(x)-f^\star(x))^2] = \mathbb{E}[g_m(x)(f(x)-f^\star(x))^2] \geqslant 0$ for any $f \in \mathcal{F}$. By Lemma 2.33, we then have $\widehat{R}_m(f^\star) \leqslant \widehat{R}_m(f) + C_\delta/2 \leqslant \widehat{R}_m(f) + \beta_m$ for any $f \in \mathcal{F}$. The elimination rule in Algorithm 2 then implies that $f^\star \in \mathcal{F}_m$ for any $m \in [M]$.

2. Fix any $f \in \mathcal{F}_m$. With Lemma 2.33, we have

$$\begin{aligned}\sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t[M_t(f)] &\leqslant 2\sum_{t=1}^{\tau_{m-1}} M_t(f) + C_\delta \\ &= 2\widehat{R}_m(f) - 2\widehat{R}_m(f^\star) + C_\delta \\ &\leqslant 2\widehat{R}_m(f) - 2\widehat{R}_m(\widehat{f}_m) + C_\delta\end{aligned}$$

$$\leqslant 2\beta_m + C_\delta,$$

where the third line comes from the fact that $\widehat{f}_m$ is the minimizer of $\widehat{R}_m(\cdot)$; and the last line comes from the fact that $f \in \mathcal{F}_m$.

3. Fix any $f \in \mathcal{F}_{m+1}$. We have

$$\begin{aligned}
\widehat{R}_m(f) - \widehat{R}_m(\widehat{f}_m) &\leqslant \widehat{R}_m(f) - \widehat{R}_m(f^\star) + \frac{C_\delta}{2} \\
&= \widehat{R}_{m+1}(f) - \widehat{R}_{m+1}(f^\star) - \sum_{t=\tau_{m-1}+1}^{\tau_m} M_t(f) + \frac{C_\delta}{2} \\
&\leqslant \widehat{R}_{m+1}(f) - \widehat{R}_{m+1}(\widehat{f}_{m+1}) - \sum_{t=\tau_{m-1}+1}^{\tau_m} \mathbb{E}_t[M_t(f)]/2 + C_\delta \\
&\leqslant \beta_{m+1} + C_\delta \\
&= \beta_m,
\end{aligned}$$

where the first line comes from Lemma 2.33; the third line comes from the fact that $\widehat{f}_{m+1}$ is the minimizer with respect to $\widehat{R}_{m+1}$ and Lemma 2.33; the last line comes from the definition of β_m.

□

Lemma 2.36. *For any* $m \in [M]$, *we have* $g_m(x) = 1 \implies w(x; \mathcal{F}_m) > \gamma$.

Proof. We only need to show that $\mathsf{ucb}(x; \mathcal{F}_m) - \mathsf{lcb}(x; \mathcal{F}_m) \leqslant \gamma \implies g_m(x) = 0$. Suppose otherwise $g_m(x) = 1$, which implies that both

$$\frac{1}{2} \in (\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)) \quad \text{and} \quad [\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)] \nsubseteq \left[\frac{1}{2} - \gamma, \frac{1}{2} + \gamma\right]. \tag{2.7}$$

If $\frac{1}{2} \in (\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m))$ and $\mathsf{ucb}(x; \mathcal{F}_m) - \mathsf{lcb}(x; \mathcal{F}_m) \leqslant \gamma$, we must have $\mathsf{lcb}(x; \mathcal{F}_m) \geqslant \frac{1}{2} - \gamma$ and $\mathsf{ucb}(x; \mathcal{F}_m) \leqslant \frac{1}{2} + \gamma$, which contradicts with Eq. (2.7). □

We introduce more notations. Fix any $m \in [M]$. We use $n_m := \tau_m - \tau_{m-1}$ to denote the length of epoch m, and use abbreviation $\rho_m := 2\beta_m + C_\delta$. Denote $(\mathcal{X}, \Sigma, \mathcal{D}_\mathcal{X})$ as the (marginal) probability space, and denote $\overline{\mathcal{X}}_m := \{x \in \mathcal{X} : g_m(x) = 1\} \in \Sigma$ be the region where query *is* requested within epoch m. Since we have $\mathcal{F}_{m+1} \subseteq \mathcal{F}_m$ by Lemma 2.35, we clearly have $\overline{\mathcal{X}}_{m+1} \subseteq \overline{\mathcal{X}}_m$. We now define a sub probability measure $\bar{\mu}_m := (\mathcal{D}_\mathcal{X})_{|\overline{\mathcal{X}}_m}$ such that $\bar{\mu}_m(\omega) = \mathcal{D}_\mathcal{X}(\omega \cap \overline{\mathcal{X}}_m)$ for any $\omega \in \Sigma$. Fix any time step t within epoch m and any $\overline{m} \leqslant m$. Consider any measurable function F (that is $\mathcal{D}_\mathcal{X}$ integrable), we have

$$\begin{aligned}
\mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) = 1) \cdot F(x)] &= \int_{x\in\overline{\mathcal{X}}_m} F(x)\, d\mathcal{D}_\mathcal{X}(x) \\
&\leqslant \int_{x\in\overline{\mathcal{X}}_{\overline{m}}} F(x)\, d\mathcal{D}_\mathcal{X}(x) \\
&= \int_{x\in\mathcal{X}} F(x)\, d\bar{\mu}_{\overline{m}}(x) \\
&=: \mathbb{E}_{x\sim\bar{\mu}_{\overline{m}}}[F(x)], \qquad (2.8)
\end{aligned}$$

where, by a slightly abuse of notations, we use $\mathbb{E}_{x\sim\mu}[\cdot]$ to denote the integration with any sub probability measure μ. In particular, Eq. (2.8) holds with equality when $\overline{m} = m$.

Lemma 2.37. *Fix any epoch $m \geqslant 2$. We have*

$$\mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) = 1)] \leqslant \frac{4\rho_m}{\tau_{m-1}\gamma^2} \cdot \theta^{\mathrm{val}}_{f^\star}\Big(\mathcal{F}, \gamma/2, \sqrt{\rho_m/2\tau_{m-1}}\Big).$$

Proof. We know that $\mathbb{1}(g_m(x) = 1) = \mathbb{1}(g_m(x) = 1) \cdot \mathbb{1}(w(x; \mathcal{F}_m) > \gamma)$ from Lemma 2.36. Thus, for any $\overline{m} \leqslant m$, we have

$$\begin{aligned}
\mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) = 1)] &= \mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) = 1) \cdot \mathbb{1}(w(x; \mathcal{F}_m) > \gamma)] \\
&\leqslant \mathbb{E}_{x\sim\bar{\mu}_{\overline{m}}}[\mathbb{1}(w(x; \mathcal{F}_m) > \gamma)] \\
&\leqslant \mathbb{E}_{x\sim\bar{\mu}_{\overline{m}}}\Big(\mathbb{1}\big(\exists f \in \mathcal{F}_m, |f(x) - f^\star(x)| > \gamma/2\big)\Big), \qquad (2.9)
\end{aligned}$$

where the second line uses Eq. (2.8) and the last line comes from the facts that $f^\star \in \mathcal{F}_m$ and $w(x;\mathcal{F}_m) > \gamma \implies \exists f \in \mathcal{F}_m, |f(x) - f^\star(x)| > \gamma/2$.

For any time step t, let $m(t)$ denote the epoch where t belongs to. From Lemma 2.35, we know that, $\forall f \in \mathcal{F}_m$,

$$\begin{aligned}
\rho_m &\geqslant \sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t\Big[Q_t\big(f(x_t) - f^\star(x_t)\big)^2\Big] \\
&= \sum_{t=1}^{\tau_{m-1}} \mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}\Big[\mathbb{1}(g_{m(t)}(x) = 1) \cdot \big(f(x) - f^\star(x)\big)^2\Big] \\
&= \sum_{\overline{m}=1}^{m-1} n_{\overline{m}} \cdot \mathbb{E}_{x\sim\bar{\mu}_{\overline{m}}}\Big[(f(x) - f^\star(x))^2\Big] \\
&= \tau_{m-1}\mathbb{E}_{x\sim\bar{\nu}_m}\Big[(f(x) - f^\star(x))^2\Big], \qquad (2.10)
\end{aligned}$$

where we use $Q_t = g_{m(t)}(x_t) = \mathbb{1}(g_{m(t)}(x) = 1)$ and Eq. (2.8) on the second line, and define a new sub probability measure

$$\bar{\nu}_m := \frac{1}{\tau_{m-1}} \sum_{\overline{m}=1}^{m-1} n_{\overline{m}} \cdot \bar{\mu}_{\overline{m}}$$

on the third line.

Plugging Eq. (2.10) into Eq. (2.9) leads to the bound

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) = 1)] \\
&\leqslant \mathbb{E}_{x\sim\bar{\nu}_m}\left[\mathbb{1}\left(\exists f \in \mathcal{F}, \big|f(x) - f^\star(x)\big| > \gamma/2, \mathbb{E}_{x\sim\bar{\nu}_m}\Big[\big(f(x) - f^\star(x)\big)^2\Big] \leqslant \frac{\rho_m}{\tau_{m-1}}\right)\right],
\end{aligned}$$

where we use the definition of $\bar{\nu}_m$ again (note that Eq. (2.9) works with any $\overline{m} \leqslant m$). Combining the above result with the discussion around Proposition 2.28 and Definition 2.6, we then have

$$\mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) = 1)] \leqslant \frac{4\rho_m}{\tau_{m-1}\gamma^2} \cdot \theta^{\text{val}}_{f^\star}\Big(\mathcal{F}, \gamma/2, \sqrt{\rho_m/2\tau_{m-1}}\Big).$$

□

Lemma 2.38. *Fix any epoch* $m \geqslant 2$. *We have*

$$\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x)=1)\cdot w(x;\mathcal{F}_m)] \leqslant \frac{4\rho_m}{\tau_{m-1}\gamma}\cdot\theta^{\mathrm{val}}_{f^\star}\Big(\mathcal{F},\gamma/2,\sqrt{\rho_m/2\tau_{m-1}}\Big).$$

Proof. Similar to the proof of Lemma 2.37, we have

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x)=1)\cdot w(x;\mathcal{F}_m)]\\
&=\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x)=1)\cdot\mathbb{1}(w(x;\mathcal{F}_m)>\gamma)\cdot w(x;\mathcal{F}_m)]\\
&\leqslant\mathbb{E}_{x\sim\bar{\mu}_{\bar{m}}}[\mathbb{1}(w(x;\mathcal{F}_m)>\gamma)\cdot w(x;\mathcal{F}_m)]
\end{aligned}$$

for any $\bar{m}\leqslant m$. With $\bar{\nu}_m=\frac{1}{\tau_{m-1}}\sum_{\bar{m}=1}^{m-1}n_{\bar{m}}\cdot\bar{\mu}_{\bar{m}}$, we then have

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x)=1)\cdot w(x;\mathcal{F}_m)]\\
&\leqslant\mathbb{E}_{x\sim\bar{\nu}_m}[\mathbb{1}(w(x;\mathcal{F}_m)>\gamma)\cdot w(x;\mathcal{F}_m)]\\
&\leqslant\mathbb{E}_{x\sim\bar{\nu}_m}\left[\mathbb{1}(\exists f\in\mathcal{F}_m,\big|f(x)-f^\star(x)\big|>\gamma/2)\cdot\left(\sup_{f,f'\in\mathcal{F}_m}|f(x)-f'(x)|\right)\right]\\
&\leqslant 2\mathbb{E}_{x\sim\bar{\nu}_m}\left[\mathbb{1}(\exists f\in\mathcal{F}_m,\big|f(x)-f^\star(x)\big|>\gamma/2)\cdot\left(\sup_{f\in\mathcal{F}_m}|f(x)-f^\star(x)|\right)\right]\\
&\leqslant 2\int_{\gamma/2}^1\mathbb{E}_{x\sim\bar{\nu}_m}\left[\mathbb{1}\left(\sup_{f\in\mathcal{F}_m}\big|f(x)-f^\star(x)\big|\geqslant\omega\right)\right]\mathrm{d}\,\omega\\
&\leqslant 2\int_{\gamma/2}^1\frac{1}{\omega^2}\,\mathrm{d}\,\omega\cdot\left(\frac{\rho_m}{\tau_{m-1}}\cdot\theta^{\mathrm{val}}_{f^\star}\Big(\mathcal{F},\gamma/2,\sqrt{\rho_m/2\tau_{m-1}}\Big)\right)\\
&\leqslant\frac{4\rho_m}{\tau_{m-1}\gamma}\cdot\theta^{\mathrm{val}}_{f^\star}\Big(\mathcal{F},\gamma/2,\sqrt{\rho_m/2\tau_{m-1}}\Big),
\end{aligned}$$

where we use similar steps as in the proof of Lemma 2.37. □

Lemma 2.39. *Fix any* $m\in[M]$. *We have* $\mathsf{excess}_\gamma(\widehat{h}_m;x)\leqslant 0$ *if* $g_m(x)=0$.

Proof. Recall that

$$\begin{aligned}\mathsf{excess}_\gamma(\widehat{h};x) = \mathbb{1}\big(\widehat{h}(x) \neq \perp\big) \cdot \big(\mathbb{P}_y\big(y \neq \mathrm{sign}(\widehat{h}(x))\big) - \mathbb{P}_y\big(y \neq \mathrm{sign}(h^\star(x))\big)\big) \\ + \mathbb{1}\big(\widehat{h}(x) = \perp\big) \cdot \big((1/2-\gamma) - \mathbb{P}_y\big(y \neq \mathrm{sign}(h^\star(x))\big)\big).\end{aligned}$$

We now analyze the event $\{g_m(x) = 0\}$ in two cases.

Case 1: $\widehat{h}_m(x) = \perp$**.**

Since $\eta(x) = f^\star(x) \in [\mathsf{lcb}(x;\mathcal{F}_m), \mathsf{ucb}(x;\mathcal{F}_m)]$, we know that $\eta(x) \in [\frac{1}{2}-\gamma, \frac{1}{2}+\gamma]$ and thus $\mathbb{P}_y\big(y \neq \mathrm{sign}(h^\star(x))\big) \geqslant \frac{1}{2}-\gamma$. As a result, we have $\mathsf{excess}_\gamma(\widehat{h}_m;x) \leqslant 0$.

Case 2: $\widehat{h}_m(x) \neq \perp$ **but** $\frac{1}{2} \notin (\mathsf{lcb}(x;\mathcal{F}_m), \mathsf{ucb}(x;\mathcal{F}_m))$**.**

In this case, we know that $\mathrm{sign}(\widehat{h}_m(x)) = \mathrm{sign}(h^\star(x))$ whenever $\eta(x) \in [\mathsf{lcb}(x;\mathcal{F}_m), \mathsf{ucb}(x;\mathcal{F}_m)]$. As a result, we have $\mathsf{excess}_\gamma(\widehat{h}_m;x) \leqslant 0$ as well. □

2.5.4 Proofs and Supporting Results for Section 2.3

Proposition 2.10. *The classifier $\widehat{h}$ returned by Algorithm 1 enjoys proper abstention. With randomization over the abstention region, we have the following upper bound on its standard excess error*

$$\mathrm{err}(\check{h}) - \mathrm{err}(h^\star) \leqslant \mathrm{err}_\gamma(\widehat{h}) - \mathrm{err}(h^\star) + \gamma \cdot \mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X}_\gamma). \tag{2.5}$$

Proof. The proper abstention property of $\widehat{h}$ returned by Algorithm 1 is achieved via conservation: $\widehat{h}$ will avoid abstention unless it is absolutely sure that abstention is the optimal choice. The proper abstention property implies that $\mathbb{P}_{x\sim\mathcal{D}_X}(\widehat{h}(x) = \perp) \leqslant \mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X}_\gamma)$. The desired result follows by combining this inequality with Eq. (2.4). □

Theorem 2.13. *With an appropriate choice of the abstention parameter γ in Algorithm 1 and randomization over the abstention region, Algorithm 1 learns a classifier $\check{h}$ at the minimax optimal rates: To achieve ε standard excess error, it takes $\widetilde{\Theta}(\tau_0^{-2})$ labels under Massart noise and takes $\widetilde{\Theta}(\varepsilon^{-2/(1+\beta)})$ labels under Tsybakov noise.*

Proof. The results follow by taking the corresponding γ in Algorithm 1 and then apply Proposition 2.10. In the case with Massart noise, we have $\mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X}_\gamma) = 0$ when $\gamma = \tau_0$; and the corresponding label complexity scales as $\widetilde{O}(\tau_0^{-2})$. In the case with Tsybakov noise, we have $\mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X}_\gamma) = \frac{\varepsilon}{2}$ when $\gamma = (\frac{\varepsilon}{2c})^{1/(1+\beta)}$. Applying Algorithm 1 to achieve $\frac{\varepsilon}{2}$ Chow's excess error thus leads to $\frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$ standard excess error. The corresponding label complexity scales as $\widetilde{O}(\varepsilon^{-2/(1+\beta)})$. □

Theorem 2.18. *With an appropriate choice of the abstention parameter γ in Algorithm 1 and randomization over the abstention region, Algorithm 1 learns a classifier $\check{h}$ with $\varepsilon + \zeta_0$ standard excess error after querying $\widetilde{\Theta}(\tau_0^{-2})$ labels under Definition 2.15 or querying $\widetilde{\Theta}(\varepsilon^{-2/(1+\beta)})$ labels under Definition 2.16.*

Proof. For any abstention parameter $\gamma > 0$, we denote $\mathcal{X}_{\zeta_0,\gamma} := \{x \in \mathcal{X} : \eta(x) \in [\frac{1}{2} - \gamma, \frac{1}{2} + \gamma], |\eta(x) - 1/2| > \zeta_0\}$ as the intersection of the region controlled by noise-seeking conditions and the (possible) abstention region. Let $\widehat{h}$ be the classifier returned by Algorithm 1 and $\check{h}$ be its randomized version (over the abstention region). We denote $\mathcal{S} := \{x \in \mathcal{X} : \widehat{h}(x) = \perp\}$ be the abstention region of $\widehat{h}$. Since $\widehat{h}$ abstains properly, we have $\mathcal{S} \subseteq \{x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \gamma\} =: \mathcal{X}_\gamma$. We write $\mathcal{S}_0 := \mathcal{S} \cap \mathcal{X}_{\zeta_0,\gamma}$, $\mathcal{S}_1 := \mathcal{S} \setminus \mathcal{S}_0$ and $\mathcal{S}_2 := \mathcal{X} \setminus \mathcal{S}$. For any $h : \mathcal{X} \to \mathcal{Y}$, we define the notation $\mathsf{excess}(h; x) := (\mathbb{P}_{y|x}(y \neq \mathrm{sign}(h(x))) - \mathbb{P}_{y|x}(y \neq \mathrm{sign}(h^\star(x))))$, and have $\mathsf{excess}(h) = \mathbb{E}_{x\sim\mathcal{D}_X}[\mathsf{excess}(h; x)]$. We then have

$$
\begin{aligned}
\mathsf{excess}(\check{h}) &= \mathbb{E}_{x\sim\mathcal{D}_X}\Big[\mathsf{excess}(\check{h}; x) \cdot \mathbb{1}(x \in \mathcal{S}_0)\Big] + \mathbb{E}_{x\sim\mathcal{D}_X}\Big[\mathsf{excess}(\check{h}; x) \cdot \mathbb{1}(x \in \mathcal{S}_1)\Big] \\
&\quad + \mathbb{E}_{x\sim\mathcal{D}_X}\Big[\mathsf{excess}(\check{h}; x) \cdot \mathbb{1}(x \in \mathcal{S}_2)\Big] \\
&\leqslant \gamma \cdot \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(x \in \mathcal{S}_0)] + \zeta_0 \cdot \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(x \in \mathcal{S}_1)] \\
&\quad + \mathbb{E}_{x\sim\mathcal{D}_X}[\mathsf{excess}_\gamma(\widehat{h}; x) \cdot \mathbb{1}(x \in \mathcal{S}_2)] \\
&\leqslant \gamma \cdot \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(x \in \mathcal{X}_{\zeta_0,\gamma})] + \zeta_0 + \varepsilon/2,
\end{aligned}
$$

where the bound on the second term comes from the fact that $\mathcal{S} \subseteq \mathcal{X}_\gamma$ and the bound on the third term comes from the same analysis that appears in the proof of Theorem 2.7 (with $\varepsilon/2$ accuracy). One can then tune γ in ways discussed in the proof

of Theorem 2.13 to bound the first term by $\varepsilon/2$, i.e., $\gamma \cdot \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(x \in \mathcal{X}_{\zeta_0,\gamma})] \leqslant \varepsilon/2$, with similar label complexity. □

Proposition 2.17. *Fix $\varepsilon, \delta, \gamma > 0$. For any labeling budget $B \gtrsim \frac{1}{\gamma^2} \cdot \log^2(\frac{1}{\varepsilon\gamma}) \cdot \log(\frac{1}{\varepsilon\gamma\delta})$, there exists a learning problem (with a set of linear regression functions) satisfying Definition 2.15/Definition 2.16 such that (1) any "uncertainty-based" active learner suffers expected standard excess error $\Omega(B^{-1})$; yet (2) with probability at least $1-\delta$, Algorithm 1 returns a classifier with standard excess error at most ε.*

Before proving Proposition 2.17, we first construct a simple problem with linear regression function and give the formal definition of "uncertainty-based" active learner.

Example 2.40. *We consider the case where $\mathcal{X} = [0,1]$ and $\mathcal{D}_X = \mathrm{unif}(\mathcal{X})$. We consider feature embedding $\phi : \mathcal{X} \to \mathbb{R}^2$, i.e., $\phi(x) = [\phi_1(x), \phi_2(x)]^\top$. We take $\phi_1(x) := 1$ for any $x \in \mathcal{X}$, and define $\phi_2(x)$ as*

$$\phi_2(x) := \begin{cases} 0, & x \in \mathcal{X}_{\mathrm{hard}}, \\ 1, & x \in \mathcal{X}_{\mathrm{easy}}, \end{cases}$$

where $\mathcal{X}_{\mathrm{easy}} \subseteq \mathcal{X}$ is any subset such that $\mathcal{D}_X(\mathcal{X}_{\mathrm{easy}}) = p$, for some constant $p \in (0,1)$, and $\mathcal{X}_{\mathrm{hard}} = \mathcal{X} \setminus \mathcal{X}_{\mathrm{easy}}$. We consider a set of linear regression function $\mathcal{F} := \{f_\theta : f_\theta(x) = \langle\phi(x), \theta\rangle, \|\theta\|_2 \leqslant 1\}$. We set $f^\star = f_{\theta^\star}$, where $\theta^\star = [\theta_1^\star, \theta_2^\star]^\top$ is selected such that $\theta_1^\star = \frac{1}{2}$ and $\theta_2^\star = \mathrm{unif}(\{\pm\frac{1}{2}\})$.

Definition 2.41. *We say an algorithm is a "uncertainty-based" active learner if, for any $x \in \mathcal{X}$, the learner*

- *constructs an open confidence interval $(\mathsf{lcb}(x), \mathsf{ucb}(x))$ with $\eta(x) \in (\mathsf{lcb}(x), \mathsf{ucb}(x))$;*[9]
- *queries the label of $x \in \mathcal{X}$ if $\frac{1}{2} \in (\mathsf{lcb}(x), \mathsf{ucb}(x))$.*

[9] By restricting to learners that construct an open confidence interval containing $\eta(x)$, we do not consider the corner cases when $\mathsf{lcb}(x) = \frac{1}{2}$ or $\mathsf{ucb}(x) = \frac{1}{2}$ and the confidence interval close.

Proof. With any given labeling budget B, we consider the problem instance described in Example 2.40 with $p = B^{-1}/2$. We can easily see that this problem instance satisfy Definition 2.15 and Definition 2.16.

We first consider any "uncertainty-based" active learner. Let Z denote the number of data points lie in $\mathcal{X}_{\text{easy}}$ among the first B random draw of examples. We see that $Z \sim \mathcal{B}(B, B^{-1}/2)$ follows a binomial distribution with B trials and $B^{-1}/2$ success rate. By Markov inequality, we have

$$\mathbb{P}\left(Z \geqslant \frac{3}{2}\mathbb{E}[Z]\right) = \mathbb{P}\left(Z \geqslant \frac{3}{4}\right) \leqslant \frac{2}{3}.$$

That being said, with probability at least $1/3$, there will be $Z = 0$ data point that randomly drawn from the easy region $\mathcal{X}_{\text{easy}}$. We denote that event as $\mathcal{E}$. Since $\eta(x) = f^\star(x) = \frac{1}{2}$ for any $x \in \mathcal{X}_{\text{hard}}$, any "uncertainty-based" active learner will query the label of any data point $x \in \mathcal{X}_{\text{hard}}$. As a result, under event $\mathcal{E}$, the active learner will use up all the labeling budget in the first B rounds and observe zero label for any data point $x \in \mathcal{X}_{\text{easy}}$. Since the easy region $\mathcal{X}_{\text{easy}}$ has measure $B^{-1}/2$ and $\theta_2^\star = \text{unif}(\{\pm\frac{1}{2}\})$, any classification rule over the easy region would results in expected excess error lower bounded by $B^{-1}/4$. To summarize, with probability at least $\frac{1}{3}$, any "uncertainty-based" active learner without abstention suffers expected excess error $\Omega(B^{-1})$.

We now consider the classifier returned by Algorithm 1.[10] For the linear function considered in Example 2.40, we have $\text{Pdim}(\mathcal{F}) \leqslant 2$ (Haussler, 1989) and $\theta^{\text{val}}_{f^\star}(\mathcal{F}, \gamma/2, \varepsilon) \leqslant 2$ for any $\varepsilon \geqslant 0$ (see Section 2.5.1). Thus, by setting $T = O(\frac{1}{\varepsilon\,\gamma} \cdot \log(\frac{1}{\varepsilon\,\gamma\,\delta}))$, with probability at least $1 - \delta$, Algorithm 1 return a classifier $\widehat{h}$ with Chow's excess error at most ε and label complexity $O(\frac{1}{\gamma^2} \cdot \log^2(\frac{1}{\varepsilon\,\gamma}) \cdot \log(\frac{1}{\varepsilon\,\gamma\,\delta})) = \text{poly}(\frac{1}{\gamma}, \log(\frac{1}{\varepsilon\,\gamma\,\delta}))$. Since $\widehat{h}$ enjoys proper abstention, it never abstains for $x \in \mathcal{X}_{\text{easy}}$. Note that we have $\eta(x) = \frac{1}{2}$ for any $x \in \mathcal{X}_{\text{hard}}$. By randomizing the prediction of $\widehat{h}$ over the abstention region, we obtain a randomized classifier with standard excess

[10] The version that works with an infinite set of regression functions using concentration results presented in Lemma 2.33. Or, one can first discretie the set of regression function and then use the version presented in Algorithm 1.

error at most ε. □

2.5.5 Proofs and Supporting Results for Section 2.4.1

We introduce a new perspective for designing and analyzing active learning algorithms in Section 2.5.5.1 (with new notations introduced). Based on this new perspective, we present our algorithm and its theoretical guarantees in Section 2.5.5.2. Supporting lemmas are deferred to Section 2.5.5.3.

2.5.5.1 The perspective: Regret Minimization with Selective Sampling

We view active learning as a decision making problem: at each round, the learner selects an action, suffers a loss (that may not be observable), and decides to query the label or not. At a high level, the learner aims at *simultaneously* minimizing the regret and the number of queries; and will randomly return a classifier/decision rule at the end of the learning process.

The perspective is inspired by the seminal results derived in Dekel et al. (2012), where the authors study active learning with linear regression functions and focus on standard excess error guarantees. With this regret minimization perspective, we can also take advantage of fruitful results developed in the field of contextual bandits (Russo and Van Roy, 2013; Foster et al., 2020c).

Decision making for regret minimization. To formulate the regret minimization problem, we consider the action set $\mathcal{A} = \{+1, -1, \perp\}$, where the action $+1$ (resp. -1) represents labeling any data point $x \in \mathcal{X}$ as positive (resp. negative); and the action $\perp$ represents abstention. At each round $t \in [T]$, the learner observes a data point $x_t \in \mathcal{X}$ (which can be chosen by an adaptive adversary), takes an action $a_t \in \mathcal{A}$, and then suffers a loss, which is defined as

$$\ell_t(a_t) = \mathbb{1}(\mathrm{sign}(y_t) \neq a_t, a_t \neq \perp) + \left(\frac{1}{2} - \gamma\right) \cdot \mathbb{1}(a_t = \perp).$$

We use $a_t^\star := \mathrm{sign}(2f^\star(x_t) - 1) = \mathrm{sign}(2\eta(x_t) - 1)$ to denote the action taken by the Bayes optimal classifier $h^\star \in \mathcal{H}$. Denote filtration $\mathfrak{F}_t := \sigma((x_i, y_i)_{i=1}^t)$. We define the (conditionally) expected regret at time step $t \in [T]$ as

$$\mathbf{Reg}_t = \mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \mathfrak{F}_{t-1}].$$

The (conditionally) expected cumulative regret across T rounds is defined as

$$\mathbf{Reg}(T) = \sum_{t=1}^{T} \mathbf{Reg}_t,$$

which is the target that the learner aims at minimizing.

Selective querying for label efficiency. Besides choosing an action $a_t \in \mathcal{A}$ at each time step, our algorithm also determines *whether or not* to query the label y_t with respect to x_t. Note that such selective querying protocol makes our problem different from contextual bandits (Russo and Van Roy, 2013; Foster et al., 2020c): The loss $\ell_t(a_t)$ of an chosen a_t may not be even observed.

We use Q_t to indicate the query status at round t, i.e.,

$$Q_t = \mathbb{1}(\text{label } y_t \text{ of } x_t \text{ is queried}).$$

The learner also aims at minimizing the total number of queries across T rounds, i.e., $\sum_{t=1}^T Q_t$.

Connection to active learning. We consider the following learner for the above mentioned decision making problem. At each round, the learner constructs a classifier $\widehat{h}_t : \mathcal{X} \to \{+1, -1, \perp\}$ and a query function $g_t : \mathcal{X} \to \{0, 1\}$; the learner then takes action $a_t = \widehat{h}_t(x_t)$ and decides the query status as $Q_t = g_t(x_t)$.

Conditioned on $\mathfrak{F}_{t-1}$, taking expectation over $\ell_t(a_t)$ leads to the following equiv-

alence.

$$
\begin{aligned}
&\mathbb{E}[\ell_t(a_t) \mid \mathfrak{F}_{t-1}] \\
&= \mathbb{E}\left[\mathbb{1}(\operatorname{sign}(y_t) \neq a_t, a_t \neq \perp) + \left(\frac{1}{2} - \gamma\right) \cdot \mathbb{1}(a_t = \perp) \mid \mathfrak{F}_{t-1}\right] \\
&= \mathbb{E}\left[\mathbb{1}\big(\operatorname{sign}(y_t) \neq \widehat{h}(x_t), \widehat{h}(x_t) \neq \perp\big) + \left(\frac{1}{2} - \gamma\right) \cdot \mathbb{1}\big(\widehat{h}(x_t) = \perp\big) \mid \mathfrak{F}_{t-1}\right] \\
&= \mathbb{P}_{(x,y)\sim\mathcal{D}_{xy}}\big(\operatorname{sign}(y) \neq \widehat{h}(x), \widehat{h}(x) \neq \perp\big) + \left(\frac{1}{2} - \gamma\right) \cdot \mathbb{P}(\widehat{h}(x) = \perp) \\
&= \operatorname{err}_\gamma(\widehat{h}_t).
\end{aligned}
$$

This shows that the (conditionally) expected instantaneous loss precisely captures the Chow's error of classifier $\widehat{h}_t$. Similarly, we have

$$
\mathbb{E}[\ell_t(a_t^\star) \mid \mathfrak{F}_{t-1}] = \mathbb{P}_{(x,y)\sim\mathcal{D}_{xy}}(\mathbb{1}(y \neq \operatorname{sign}(2\eta(x) - 1))) = \operatorname{err}(h^\star).
$$

Combining the above two results, we notice that the (conditionally) expected instantaneous *regret* exactly captures the Chow's excess error of classifier $\widehat{h}_t$, i.e.,

$$
\mathbf{Reg}_t = \operatorname{err}_\gamma(\widehat{h}_t) - \operatorname{err}(h^\star).
$$

Let $\widehat{h} \sim \operatorname{unif}(\{\widehat{h}_t\}_{t=1}^T)$ be a classifier randomly selected from all the constructed classifiers. Taking expectation with respect to this random selection procedure, we then have

$$
\mathbb{E}_{\widehat{h}\sim\operatorname{unif}(\{\widehat{h}_t\}_{t=1}^T)}[\operatorname{err}_\gamma(\widehat{h}) - \operatorname{err}(h^\star)] = \sum_{t=1}^{T} (\operatorname{err}_\gamma(\widehat{h}_t) - \operatorname{err}(h^\star))/T = \mathbf{Reg}(T)/T. \tag{2.11}
$$

That being said, the expected Chow's excess error of $\widehat{h}$ can be sublinear in T. If the total number of queries is logarithmic in T, this immediately implies learning a classifier with exponential savings in label complexity.

2.5.5.2 Algorithm and Main Results

We present an algorithm that achieves constant label complexity next (Algorithm 2). Compared to Algorithm 1, Algorithm 2 drops the epoch scheduling, uses a sharper elimination rule for the active set (note that β doesn't depend on T, due to applying optimal stopping theorem in Lemma 2.44), and is analyzed with respect to eluder dimension (Definition 2.24) instead of disagreement coefficient. As a result, we shave all three sources of $\log \frac{1}{\varepsilon}$, and achieve constant label complexity for general $\mathcal{F}$ (as long as it's finite and has finite eluder dimension). We abbreviate $\mathfrak{e} := \sup_{f^\star \in \mathcal{F}} \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2)$.

Algorithm 2 Efficient Active Learning with Abstention (Constant Label Complexity)

Input: Time horizon $T \in \mathbb{N}$, abstention parameter $\gamma \in (0, 1/2)$ and confidence level $\delta \in (0,1)$.

Initialize $\widehat{\mathcal{H}} := \emptyset$. Set $T := O(\frac{e}{\varepsilon\gamma} \cdot \log(\frac{|\mathcal{F}|}{\delta}))$ and $\beta := \frac{1}{2}\log(\frac{|\mathcal{F}|}{\delta})$.

for $t = 1, 2, \ldots, T$ **do**

Get $\widehat{f}_t := \arg\min_{f \in \mathcal{F}} \sum_{i<t} Q_i (f(x_i) - y_i)^2$.

`//We use` $Q_t \in \{0,1\}$ `to indicate whether the label of` x_t `is queried.`

(Implicitly) Construct active set of regression function $\mathcal{F}_t \subseteq \mathcal{F}$ as

$$\mathcal{F}_t := \left\{ f \in \mathcal{F} : \sum_{i=1}^{t-1} Q_i (f(x_i) - y_i)^2 \leqslant \sum_{i=1}^{t-1} Q_i (\widehat{f}_t(x_i) - y_i)^2 + \beta \right\}.$$

Construct classifier $\widehat{h}_t : \mathcal{X} \to \{+1, -1, \perp\}$ as

$$\widehat{h}_t(x) := \begin{cases} \perp, & \text{if } [\mathsf{lcb}(x; \mathcal{F}_t), \mathsf{ucb}(x; \mathcal{F}_t)] \subseteq \left[\frac{1}{2} - \gamma, \frac{1}{2} + \gamma\right]; \\ \mathrm{sign}(2\widehat{f}_t(x) - 1), & \text{o.w.} \end{cases}$$

Update $\widehat{\mathcal{H}} = \widehat{\mathcal{H}} \cup \{\widehat{h}_t\}$. Construct query function $g_m : \mathcal{X} \to \{0,1\}$ as

$$g_t(x) := \mathbb{1}\left(\frac{1}{2} \in (\mathsf{lcb}(x; \mathcal{F}_t), \mathsf{ucb}(x; \mathcal{F}_t))\right) \cdot \mathbb{1}(\widehat{h}_t(x) \neq \perp).$$

Observe $x_t \sim \mathcal{D}_X$. Take action $a_t := \widehat{h}_t(x_t)$. Set $Q_t := g_t(x_t)$.

if $Q_t = 1$ **then**

Query the label y_t of x_t.

Return $\widehat{h} := \mathrm{unif}(\widehat{\mathcal{H}})$.

Before proving Theorem 2.19. We define some notations that are specialized to Section 2.5.5.

We define filtrations $\mathfrak{F}_{t-1} := \sigma(x_1, y_1, \ldots, x_{t-1}, y_{t-1})$ and $\overline{\mathfrak{F}}_{t-1} := \sigma(x_1, y_1, \ldots, x_t)$. Note that we additionally include the data point x_t in the filtration $\overline{\mathfrak{F}}_{t-1}$ at time step $t-1$. We denote $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot \mid \overline{\mathfrak{F}}_{t-1}]$. For any $t \in [T]$, we denote $M_t(f) := Q_t((f(x_t) - y_t)^2 - (f^\star(x_t) - y_t)^2)$. We have $\sum_{i=1}^{\tau} \mathbb{E}_t[M_t(f)] = \sum_{t=1}^{\tau} Q_t (f(x_t) - f^\star(x_t))^2$. For any

given data point $x_t \in \mathcal{X}$, we use abbreviations

$$\mathsf{ucb}_t := \mathsf{ucb}(x_t; \mathcal{F}_t) = \sup_{f \in \mathcal{F}_t} f(x_t) \quad \text{and} \quad \mathsf{lcb}_t := \mathsf{lcb}(x_t; \mathcal{F}_t) = \inf_{f \in \mathcal{F}_t} f(x_t)$$

to denote the upper and lower confidence bounds of $\eta(x_t) = f^\star(x_t)$. We also denote

$$w_t := \mathsf{ucb}_t - \mathsf{lcb}_t = \sup_{f, f' \in \mathcal{F}_t} |f(x_t) - f'(x_t)|$$

as the width of confidence interval.

Theorem 2.19. *With probability at least* $1 - 2\delta$, *Algorithm 2 returns a classifier with expected Chow's excess error at most* ε *and label complexity* $O(\frac{\mathfrak{e} \cdot \log(|\mathcal{F}|/\delta)}{\gamma^2})$, *which is independent of* $\frac{1}{\varepsilon}$.

Proof. We first analyze the label complexity of Algorithm 2. Note that Algorithm 2 constructs $\widehat{h}_t$ and g_t in forms similar to the ones constructed in Algorithm 1, and Lemma 2.36 holds for Algorithm 2 as well. Based on Lemma 2.36, we have $Q_t = g_t(x_t) = 1 \implies w_t > \gamma$. Thus, taking $\zeta = \gamma$ in Lemma 2.47 leads to

$$\sum_{t=1}^{T} \mathbb{1}(Q_t = 1) < \frac{17 \log(2|\mathcal{F}|/\delta)}{2\gamma^2} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2),$$

with probability one. The label complexity of Algorithm 2 is then upper bounded by a constant as long as $\mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2)$ is upper bounded by a constant (which has no dependence on T or $\frac{1}{\varepsilon}$).

We next analyze the excess error of $\widehat{h}$. We consider the good event $\mathcal{E}$ defined in Lemma 2.46, which holds true with probability at least $1 - \delta$. Under event $\mathcal{E}$, Lemma 2.50 shows that

$$\sum_{t=1}^{T} \mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \overline{\mathfrak{F}}_{t-1}] \leqslant \frac{17\sqrt{2}\beta}{\gamma} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2).$$

Since

$$\mathbb{E}\Big[\mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \bar{\mathfrak{F}}_{t-1}] \mid \mathfrak{F}_{t-1}\Big] = \mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \mathfrak{F}_{t-1}],$$

and $|\mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \bar{\mathfrak{F}}_{t-1}]| \leqslant 1$ by construction, applying Lemma 2.30 with respect to $\mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \bar{\mathfrak{F}}_{t-1}]$ further leads to

$$\mathbf{Reg}(T) = \sum_{t=1}^{T} \mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \mathfrak{F}_{t-1}] \leqslant \frac{34\sqrt{2}\beta}{\gamma} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2) + 8\log(2\delta^{-1}),$$

with probability at least $1 - 2\delta$ (due to the additional application of Lemma 2.30). Since $\widehat{h} \sim \mathrm{unif}(\widehat{\mathcal{H}})$, based on Eq. (2.11), we thus know that

$$\begin{aligned}\mathbb{E}_{\widehat{h}\sim\mathrm{unif}(\widehat{\mathcal{H}})}[\mathrm{err}_\gamma(\widehat{h}) - \mathrm{err}(h^\star)] &= \sum_{t=1}^{T} \big(\mathrm{err}_\gamma(\widehat{h}_t) - \mathrm{err}(h^\star)\big)/T \\ &\leqslant \left(\frac{34\sqrt{2}\beta}{\gamma} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2) + 8\log\big(2\delta^{-1}\big)\right)/T\end{aligned}$$

Since $T := O(\frac{\epsilon}{\varepsilon\,\gamma} \cdot \log(\frac{|\mathcal{F}|}{\delta}))$, we then know that the expected Chow's excess error is at most ε. □

Theorem 2.42. *Consider the setting where the data points $\{x_t\}_{t=1}^T$ are chosen by an adaptive adversary with $y_t \sim \mathcal{D}_{y|x_t}$. With probability at least $1 - \delta$, Algorithm 2 simultaneously guarantees*

$$\sum_{t=1}^{T} \mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \bar{\mathfrak{F}}_{t-1}] \leqslant \frac{34\sqrt{2}\beta}{\gamma} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2),$$

and

$$\sum_{t=1}^{T} \mathbb{1}(Q_t = 1) < \frac{17\log(2|\mathcal{F}|/\delta)}{2\gamma^2} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2).$$

Proof. The proof follows the same analysis as in the first part of the proof of Theorem 2.19 (simply stopped at the step with conditioning on $\bar{\mathfrak{F}}_{t-1}$). □

We redefine $\mathfrak{e} := \sup_{f^\star \in \mathcal{F}} \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/4)$ in the following Theorem 2.43 to account for the induced approximation error in efficient implementation.

Theorem 2.43. *Algorithm 2 can be efficiently implemented via the regression oracle and enjoys the same theoretical guarantees stated in Theorem 2.19 or Theorem 2.42. The number of oracle calls needed is* $O(\frac{\mathfrak{e}}{\varepsilon\,\gamma^3} \cdot \log(\frac{|\mathcal{F}|}{\delta}) \cdot \log(\frac{1}{\gamma}))$ *for a general set of regression functions* $\mathcal{F}$*, and* $O(\frac{\mathfrak{e}}{\varepsilon\,\gamma} \cdot \log(\frac{|\mathcal{F}|}{\delta}) \cdot \log(\frac{1}{\gamma}))$ *when* $\mathcal{F}$ *is convex and closed under pointwise convergence. The per-example inference time of the learned* $\widehat{h}_M$ *is* $O(\frac{1}{\gamma^2} \log \frac{1}{\gamma})$ *for general* $\mathcal{F}$*, and* $O(\log \frac{1}{\gamma})$ *when* $\mathcal{F}$ *is convex and closed under pointwise convergence.*

Proof. Denote $\mathcal{B}_t := \{(x_i, Q_i, y_i)\}_{i=1}^{\tau_{t-1}}$ At any time step $t \in [T]$ of Algorithm 2, we construct classifier $\widehat{h}_t$ and query function g_t with approximated confidence bounds, i.e.,

$$\widehat{\mathsf{lcb}}(x; \mathcal{F}_t) := \mathbf{Alg}_{\mathsf{lcb}}(x; \mathcal{B}_t, \beta_t, \alpha) \quad \text{and} \quad \widehat{\mathsf{ucb}}(x; \mathcal{F}_t) := \mathbf{Alg}_{\mathsf{ucb}}(x; \mathcal{B}_t, \beta_t, \alpha),$$

where $\mathbf{Alg}_{\mathsf{lcb}}$ and $\mathbf{Alg}_{\mathsf{ucb}}$ are subroutines discussed in Proposition 2.34 and $\alpha := \frac{\gamma}{4}$.

Since the theoretical analysis of Theorem 2.19 and Theorem 2.42 do not require an non-increasing (with respect to time step t) sampling region, i.e., $\{x \in \mathcal{X} : g_t(x) = 1\}$, we only need to approximate the confidence intervals at $\frac{\gamma}{4}$ level. This slightly save the computational complexity compared to Theorem 2.8, which approximates the confidence interval at $\frac{\gamma}{4\lceil \log T \rceil}$ level. The rest of the analysis of computational complexity follows similar steps in the proof of Theorem 2.8. □

2.5.5.3 Supporting Lemmas

Consider a sequence of random variables $(Z_t)_{t \in \mathbb{N}}$ adapted to filtration $\overline{\mathfrak{F}}_t$. We assume that $\mathbb{E}[\exp(\lambda Z_t)] < \infty$ for any λ and $\mu_t := \mathbb{E}\big[Z_t \mid \overline{\mathfrak{F}}_{t-1}\big]$. We also denote

$$\psi_t(\lambda) := \log \mathbb{E}\big[\exp(\lambda \cdot (Z_t - \mu_t)) \mid \overline{\mathfrak{F}}_{t-1}\big].$$

Lemma 2.44 (Russo and Van Roy (2013))**.** *With notations defined above. For any* $\lambda \geqslant 0$ *and* $\delta > 0$*, we have*

$$\mathbb{P}\left(\forall \tau \in \mathbb{N}, \sum_{t=1}^{\tau} \lambda Z_t \leqslant \sum_{t=1}^{\tau} (\lambda \mu_t + \psi_t(\lambda)) + \log\left(\frac{1}{\delta}\right)\right) \geqslant 1 - \delta. \tag{2.12}$$

Lemma 2.45. *Fix any* $\delta \in (0,1)$*. For any* $\tau \in [T]$*, with probability at least* $1-\delta$*, we have*

$$\sum_{t=1}^{\tau} M_t(f) \leqslant \sum_{t=1}^{\tau} \frac{3}{2}\mathbb{E}_t[M_t(f)] + C_\delta,$$

and

$$\sum_{t=1}^{\tau} \mathbb{E}_t[M_t(f)] \leqslant 2\sum_{t=1}^{\tau} M_t(f) + C_\delta,$$

where $C_\delta := \log\left(\frac{2|\mathcal{F}|}{\delta}\right)$.

Proof. Fix any $f \in \mathcal{F}$. We take $Z_t = M_t(f)$ in Lemma 2.44. We can rewrite

$$Z_t = Q_t\big((f(x_t) - f^\star(x_t))^2 + 2(f(x_t) - f^\star(x_t))\varepsilon_t\big),$$

where we use the notation $\varepsilon_t := f^\star(x_t) - y_t$. Since $\mathbb{E}_t[\varepsilon_t] = 0$ and $\mathbb{E}_t[\exp(\lambda\varepsilon_t) \mid \overline{\mathfrak{F}}_{t-1}] \leqslant \exp(\frac{\lambda^2}{8})$ a.s. by assumption, we have

$$\mu_t = \mathbb{E}_t[Z_t] = Q_t(f(x_t) - f^\star(x_t))^2,$$

and

$$\begin{aligned}
\psi_t(\lambda) &= \log \mathbb{E}\big[\exp(\lambda \cdot (Z_t - \mu_t)) \mid \overline{\mathfrak{F}}_{t-1}\big] \\
&= \log \mathbb{E}_t[\exp(2\lambda Q_t(f(x_t) - f^\star(x_t) \cdot \varepsilon_t))] \\
&\leqslant \frac{\lambda^2 (Q_t(f(x_t) - f^\star(x_t))^2}{2}
\end{aligned}$$

$$= \frac{\lambda^2 \mu_t}{2},$$

where the last line comes from the fact that $Q_t \in \{0, 1\}$. We can similarly upper bound $\mathbb{E}[\exp(\lambda Z_t)] = \mathbb{E}[\mathbb{E}_t[\exp(\lambda Z_t)]] \leqslant \exp(\lambda + \frac{\lambda^2}{2})$ by noticing the range fact that $\mu_t \leqslant 1$.

Plugging the above results into Lemma 2.44 with $\lambda = 1$ leads to

$$\sum_{t=1}^{\tau} M_t(f) \leqslant \sum_{t=1}^{\tau} \frac{3}{2} \mathbb{E}_t[M_t(f)] + \log \delta^{-1}.$$

Following the same procedures above with $Z_t = -M_t(f)$ and $\lambda = 1$ leads to

$$\sum_{t=1}^{\tau} \frac{3}{2} \mathbb{E}_t[M_t(f)] \leqslant 2 \sum_{t=1}^{\tau} M_t(f) + \log \delta^{-1}.$$

The final guarantees comes from taking a union abound over $f \in \mathcal{F}$ and splitting the probability for both directions. □

We use $\mathcal{E}$ to denote the good event considered in Lemma 2.45, we use it through out the rest of this section.

Lemma 2.46. *With probability at least* $1 - \delta$*, the followings hold true:*

1. $f^\star \in \mathcal{F}_t$ *for any* $t \in [T]$.
2. $\sum_{t=1}^{\tau-1} \mathbb{E}_t[M_t(f)] \leqslant 2C_\delta$ *for any* $f \in \mathcal{F}_\tau$.

Proof. The first statement immediately follows from Lemma 2.45 (the second inequality) and the fact that $\beta := C_\delta/2$ in Algorithm 2.

For any $f \in \mathcal{F}_\tau$, we have

$$\begin{aligned}
\sum_{t=1}^{\tau-1} \mathbb{E}_t[M_t(f)] &\leqslant 2 \sum_{t=1}^{\tau-1} Q_t\big((f(x_t) - y_t)^2 - (f^\star(x_t) - y_t)^2\big) + C_\delta \\
&\leqslant 2 \sum_{t=1}^{\tau-1} Q_t\Big((f(x_t) - y_t)^2 - (\widehat{f}_\tau(x_t) - y_t)^2\Big) + C_\delta
\end{aligned}$$

$$\leqslant 2C_\delta, \tag{2.13}$$

where the first line comes from Lemma 2.45, the second line comes from the fact that $\widehat{f}_\tau$ is the minimize among $\mathcal{F}_\tau$, and the third line comes from the fact that $f \in \mathcal{F}_\tau$ and $2\beta = C_\delta$. □

Lemma 2.47. *For any $\zeta > 0$, with probability 1, we have*

$$\sum_{t=1}^{T} \mathbb{1}(Q_t = 1) \cdot \mathbb{1}(w_t > \zeta) < \left(\frac{16\beta}{\zeta^2} + 1\right) \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \zeta/2).$$

Remark 2.48. *Similar upper bound has been established in the contextual bandit settings for $\sum_{t=1}^{T} \mathbb{1}(w_t > \zeta)$ (Russo and Van Roy, 2013; Foster et al., 2020c). Our results is established with an additional $\mathbb{1}(Q_t = 1)$ term due to selective querying in active learning.*

Proof. We give some definitions first. We say that x is ζ-independent of a sequence $x_1, \ldots, x_\tau$ if there exists a $f \in \mathcal{F}$ such that $|f(x) - f^\star(x)| > \zeta$ and $\sum_{i \leqslant \tau}(f(x_i) - f^\star(x_i))^2 \leqslant \zeta^2$. We say that x is ζ-dependent of $x_1, \ldots, x_\tau$ if we have $|f(x) - f^\star(x)| \leqslant \zeta$ for all $f \in \mathcal{F}$ such that $\sum_{i \leqslant \tau}(f(x_i) - f^\star(x_i))^2 \leqslant \zeta^2$. The eluder dimension $\check{\mathfrak{e}}_{f^\star}(\mathcal{F}, \zeta)$ can be equivalently defined as the length of the longest sequence $x_1, \ldots, x_\tau$ such that each x_i is ζ-independent of all its predecessors.

For any $t \in [T]$, and we denote $\mathcal{S}_t = \{x_i : Q_i = g_i(x_i) = 1, i \in [t]\}$ as the *queried* data points up to time step t. We assume that $|\mathcal{S}_t| = \tau$ and denote $\mathcal{S}_t = (x_{g(1)}, \ldots, x_{g(\tau)})$, where $g(i)$ represents the time step where the i-th *queried* data point is queried.

Claim 1. For any $j \in [\tau]$, $x_{g(j)}$ is $\frac{\zeta}{2}$-dependent on at most $\frac{16\beta}{\zeta^2}$ disjoint subsequences of $x_{g(1)}, \ldots, x_{g(j-1)}$.

For any $x_{g(j)} \in \mathcal{S}_t$, recall that

$$w_{g(j)} = \mathsf{ucb}_{g(j)} - \mathsf{lcb}_{g(j)} = \max_{f, f' \in \mathcal{F}_{g(j)}} |f(x_t) - f'(x_t)|.$$

If $m_{g(j)} > \zeta$, there must exists a $f \in \mathcal{F}_{g(j)}$ such that $\left|f(x_{g(j)}) - f^\star(x_{g(j)})\right| > \frac{\zeta}{2}$. Focus on this specific $f \in \mathcal{F}_{g(j)} \subseteq \mathcal{F}$. If $x_{g(j)}$ is $\frac{\zeta}{2}$-dependent on a subsequence

$x_{g(i_1)}, \dots, x_{g(i_m)}$ (of $x_{g(1)}, \dots, x_{g(j-1)}$), we must have

$$\sum_{k \leqslant m} (f(x_{g(i_k)}) - f^\star(x_{g(i_k)}))^2 > \frac{\zeta^2}{4}.$$

Suppose $x_{g(j)}$ is $\frac{\zeta}{2}$-dependent on K *disjoint* subsequences of $x_{g(1)}, \dots, x_{g(j-1)}$, according to Lemma 2.46, we must have

$$K \cdot \frac{\zeta^2}{4} < \sum_{i<j} (f(x_{g(i)}) - f^\star(x_{g(i)}))^2 = \sum_{k<g(j)} Q_k (f(x_k) - f^\star(x_k))^2 \leqslant 4\beta,$$

which implies that $K < \frac{16\beta}{\zeta^2}$.

Claim 2. Denote $d := \check{\mathfrak{e}}_{f^\star}(\mathcal{F}, \zeta/2) \geqslant 1$ and $K = \left\lfloor \frac{\tau-1}{d} \right\rfloor$. There must exists a $j \in [\tau]$ such that $x_{g(j)}$ is $\frac{\zeta}{2}$-dependent on at least K disjoint subsequences of $x_{g(1)}, \dots, x_{g(j-1)}$.

We initialize K subsequences $\mathcal{C}_i = \{x_{g(i)}\}$. If $x_{g(K+1)}$ is $\frac{\zeta}{2}$-dependent on each $\mathcal{C}_i$, we are done. If not, select a subsequence $\mathcal{C}_i$ such that $x_{g(K+1)}$ is $\frac{\zeta}{2}$-independent of and add $x_{g(K+1)}$ into this subsequence. Repeat this procedure with $j > K+1$ until $x_{g(j)}$ is $\frac{\zeta}{2}$-dependent of all $\mathcal{C}_i$ or $j = \tau$. In the later case, we have $\sum_{i \leqslant K} |\mathcal{C}_i| = \tau - 1 \geqslant Kd$. Since $|\mathcal{C}_i| \leqslant d$ by definition, we must have $|\mathcal{C}_i| = d$ for all $i \in [K]$. As a result, $x_{g(\tau)}$ must be $\frac{\zeta}{2}$-dependent of all $\mathcal{C}_i$.

It's easy to check that $\left\lfloor \frac{\tau-1}{d} \right\rfloor \geqslant \frac{\tau}{d} - 1$. Combining Claim 1 and 2, we have

$$\frac{\tau}{d} - 1 \leqslant \left\lfloor \frac{\tau - 1}{d} \right\rfloor \leqslant K < \frac{16\beta}{\zeta^2}.$$

Rearranging leads to the desired result. □

The following Lemma 2.49 is a restatement of Lemma 2.39 in the regret minimization setting.

Lemma 2.49. *If* $Q_t = 0$, *we have* $\mathbb{E}\left[\ell_t(a_t) - \ell_t(a_t^\star) \mid \overline{\mathfrak{F}}_{t-1}\right] \leqslant 0$.

Proof. Recall we have $a_t = \widehat{h}_t(x_t)$. We then have

$$\mathbb{E}\left[\ell_t(a_t) - \ell_t(a_t^\star) \mid \overline{\mathfrak{F}}_{t-1}\right]$$

$$
\begin{aligned}
&= \mathbb{P}_{y_t|x_t}\big(y_t \neq \operatorname{sign}(\widehat{h}_t(x_t))\big) \cdot \mathbb{1}\big(\widehat{h}_t(x_t) \neq \perp\big) + \big(1/2 - \gamma\big) \cdot \mathbb{1}\big(\widehat{h}_t(x_t) = \perp\big) \\
&\quad - \mathbb{P}_{y_t|x_t}\big(y_t \neq \operatorname{sign}(h^\star(x_t))\big) \\
&= \mathbb{1}\big(\widehat{h}_t(x_t) \neq \perp\big) \cdot \big(\mathbb{P}_{y_t|x_t}\big(y_t \neq \operatorname{sign}(\widehat{h}_t(x_t))\big) - \mathbb{P}_{y_t|x_t}\big(y_t \neq \operatorname{sign}(h^\star(x_t))\big)\big) \\
&\quad + \mathbb{1}\big(\widehat{h}_t(x_t) = \perp\big) \cdot \big(\big(1/2 - \gamma\big) - \mathbb{P}_{y_t|x_t}\big(y_t \neq \operatorname{sign}(h^\star(x_t))\big)\big).
\end{aligned}
$$

We now analyze the event $\{Q_t = 0\}$ in two cases.

Case 1: $\widehat{h}_t(x_t) = \perp$**.**

Since $\eta(x_t) = f^\star(x_t) \in [\mathsf{lcb}_t, \mathsf{ucb}_t]$, we further know that $\eta(x_t) \in [\frac{1}{2} - \gamma, \frac{1}{2} + \gamma]$ and thus $\mathbb{P}_{y_t|x_t}\big(y_t \neq \operatorname{sign}(h^\star(x_t))\big) \geqslant \frac{1}{2} - \gamma$. As a result, we have $\mathbb{E}\big[\ell_t(a_t) - \ell_t(a_t^\star) \mid \overline{\mathfrak{F}}_{t-1}\big] \leqslant 0$.

Case 2: $\widehat{h}_t(x_t) \neq \perp$ **but** $\frac{1}{2} \notin (\mathsf{lcb}_t, \mathsf{ucb}_t)$**.**

In this case, we know that $\operatorname{sign}(\widehat{h}_t(x_t)) = \operatorname{sign}(h^\star(x_t))$ whenever $\eta(x_t) \in [\mathsf{lcb}_t, \mathsf{ucb}_t]$. As a result, we have $\mathbb{E}\big[\ell_t(a_t) - \ell_t(a_t^\star) \mid \overline{\mathfrak{F}}_{t-1}\big] = 0$. □

Lemma 2.50. *Assume* $\mu(x_t) \in [\mathsf{lcb}_t, \mathsf{ucb}_t]$ *and* $f^\star$ *is not eliminated across all* $t \in [T]$*. We have*

$$
\sum_{t=1}^{T} \mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \overline{\mathfrak{F}}_{t-1}] \leqslant \frac{17\sqrt{2}\beta}{\gamma} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2). \tag{2.14}
$$

Proof. Lemma 2.49 shows that non-positive conditional regret is incurred at whenever $Q_t = 0$, we then have

$$
\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}[\ell_t(a_t) - \ell_t(a_t^\star) \mid \overline{\mathfrak{F}}_{t-1}] &\leqslant \sum_{t=1}^{T} \mathbb{1}(Q_t = 1)\mathbb{E}\big[\ell_t(a_t) - \ell_t(a_t^\star) \mid \overline{\mathfrak{F}}_{t-1}\big] \\
&= \sum_{t=1}^{T} \mathbb{1}(Q_t = 1) \cdot \mathbb{1}(w_t > \gamma) \cdot |2f^\star(x_t) - 1| \\
&\leqslant \sum_{t=1}^{T} \mathbb{1}(Q_t = 1) \cdot \mathbb{1}(w_t > \gamma) \cdot 2w_t,
\end{aligned}
$$

where we use Lemma 2.36 and Lemma 2.49 on the second line; and the last line comes from the fact that $|f^\star - \frac{1}{2}| \leqslant w_t$ whenever $f^\star$ is not eliminated and a query is

issued. We can directly apply $w_t \leqslant 1$ and Lemma 2.47 to bound the above terms by $\widetilde{O}(\frac{\mathfrak{e}_{f^\star}(\mathcal{F},\gamma/2)}{\gamma^2})$, which has slightly worse dependence on γ. Following Foster et al. (2020c), we take a slightly tighter analysis below.

Let $\mathcal{S}_T := \{x_i : Q_i = 1, i \in [T]\}$ denote the set of queried data points. Suppose $|\mathcal{S}_T| = \tau$. Let $i_1, \ldots, i_\tau$ be a reordering of indices within $\mathcal{S}_T$ such that $w_{i_1}(x_{i_1}) \geqslant w_{i_2}(x_{i_2}) \geqslant \ldots \geqslant w_{i_\tau}(x_{i_\tau})$. Consider any index $t \in [\tau]$ such that $w_{i_t}(x_{i_t}) \geqslant \gamma$. For any $\zeta \geqslant \gamma$, Lemma 2.47 implies that

$$t \leqslant \sum_{t=1}^{T} \mathbb{1}(Q_t = 1) \cdot \mathbb{1}(w_t(x_t) > \zeta) \leqslant \frac{17\beta}{\zeta^2} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \zeta/2) \leqslant \frac{17\beta}{\zeta^2} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2). \quad (2.15)$$

Taking $\zeta = w_{i_t}(x_{i_t})$ in Eq. (2.15) leads to the fact that

$$w_{i_t}(x_{i_t}) \leqslant \sqrt{\frac{17\beta \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2)}{t}}.$$

Taking $\zeta = \gamma$ in Eq. (2.15) leads to the fact that

$$\tau \leqslant \frac{17\beta}{\gamma^2} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2).$$

We now have

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{1}(Q_t = 1) \cdot \mathbb{1}(w_t > \gamma) \cdot 2w_t = \sum_{t=1}^{\tau} \mathbb{1}(w_{i_t} > \gamma) \cdot 2w_{i_t}(x_{i_t}) \\
&\leqslant 2 \sum_{t=1}^{\tau} \sqrt{\frac{17\beta \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2)}{t}} \\
&\leqslant \sqrt{34\beta \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2) \cdot \tau} \\
&\leqslant \frac{17\sqrt{2}\beta}{\gamma} \cdot \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma/2).
\end{aligned}$$

□

2.5.6 Proofs and Supporting Results for Section 2.4.2

2.5.6.1 Algorithm and Main Results

Algorithm 3 Efficient Active Learning with Abstention under Misspecification

Input: Accuracy level $\varepsilon > 0$, abstention parameter $\gamma \in (\varepsilon, 1/2)$ and confidence level $\delta \in (0,1)$.

Define $T := \frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}$, $M := \lceil \log_2 T \rceil$ and $C_\delta := O(\mathrm{Pdim}(\mathcal{F}) \cdot \log(T/\delta))$.

Define $\tau_m := 2^m$ for $m \geqslant 1$, $\tau_0 = 0$ and $\beta_m := (M - m + 1) \cdot \left(2\varepsilon^2 \tau_{M-1} + 2C_\delta\right)$.

for epoch $m = 1, 2, \ldots, M$ **do**

  Get $\widehat{f}_m := \arg\min_{f \in \mathcal{F}} \sum_{t=1}^{\tau_{m-1}} Q_t (f(x_t) - y_t)^2$.

  `//We use` $Q_t \in \{0,1\}$ `to indicate whether the label of` x_t `is queried.`

  (Implicitly) Construct active set of regression function $\mathcal{F}_m \subseteq \mathcal{F}$ as

$$\mathcal{F}_m := \left\{ f \in \mathcal{F} : \sum_{t=1}^{\tau_{m-1}} Q_t (f(x_t) - y_t)^2 \leqslant \sum_{t=1}^{\tau_{m-1}} Q_t (\widehat{f}_m(x_t) - y_t)^2 + \beta_m \right\}.$$

  Construct classifier $\widehat{h}_m : \mathcal{X} \to \{+1, -1, \perp\}$ as

$$\widehat{h}_m(x) := \begin{cases} \perp, & \text{if } [\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)] \subseteq \left[\frac{1}{2} - \gamma, \frac{1}{2} + \gamma\right]; \\ \mathrm{sign}(2\widehat{f}_m(x) - 1), & \text{o.w.} \end{cases}$$

  and query function $g_m : \mathcal{X} \to \{0,1\}$ as

$$g_m(x) := \mathbb{1}\left(\frac{1}{2} \in (\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m))\right) \cdot \mathbb{1}(\widehat{h}_m(x) \neq \perp).$$

  if epoch $m = M$ **then**

    Return classifier $\widehat{h}_M$.

  for time $t = \tau_{m-1} + 1, \ldots, \tau_m$ **do**

    Observe $x_t \sim \mathcal{D}_X$. Set $Q_t := g_m(x_t)$.

    if $Q_t = 1$ **then**

      Query the label y_t of x_t.

Algorithm 3 achieves the guarantees stated in Theorem 2.22. Theorem 2.22 is proved based on supporting lemmas derived in Section 2.5.6.2. Note that, under

the condition $\kappa \leqslant \varepsilon$, we still compete against the Bayes classifier $h^\star = h_{f^\star}$ in the analysis of Chow's excess error Eq. (2.2).

Theorem 2.22. *Suppose $\kappa \leqslant \varepsilon$. With probability at least $1-2\delta$, Algorithm 3 returns a classifier with Chow's excess error $O(\varepsilon \cdot \bar{\theta} \cdot \log(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}))$ and label complexity $O(\frac{\bar{\theta}\,\mathrm{Pdim}(\mathcal{F})}{\gamma^2} \cdot \log^2(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}) \cdot \log(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}))$.*

Proof. We analyze under the good event $\mathcal{E}$ defined in Lemma 2.33, which holds with probability at least $1-\delta$. Note that all supporting lemmas stated in Section 2.5.6.2 hold true under this event.

We analyze the Chow's excess error of $\widehat{h}_m$, which is measurable with respect to $\mathfrak{F}_{\tau_{m-1}}$. For any $x \in \mathcal{X}$, if $g_m(x) = 0$, Lemma 2.56 implies that $\mathrm{excess}_\gamma(\widehat{h}_m; x) \leqslant 2\kappa$. If $g_m(x) = 1$, we know that $\widehat{h}_m(x) \neq \perp$ and $\frac{1}{2} \in (\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m))$. Since $\bar{f} \in \mathcal{F}_m$ by Lemma 2.53 and $\sup_{x\in\mathcal{X}} |\bar{f}(x) - f^\star(x)| \leqslant \kappa$ by assumption. The error incurred in this case is upper bounded by

$$\begin{aligned}
\mathrm{excess}_\gamma(\widehat{h}_m; x) &\leqslant 2|f^\star(x) - 1/2| \\
&\leqslant 2\kappa + 2|\bar{f}(x) - 1/2| \\
&\leqslant 2\kappa + 2w(x; \mathcal{F}_m).
\end{aligned}$$

Combining these two cases together, we have

$$\mathrm{excess}_\gamma(\widehat{h}_m) \leqslant 2\kappa + 2\mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) = 1) \cdot w(x; \mathcal{F}_m)].$$

Take $m = M$ and apply Lemma 2.55 leads to the following guarantee.

$$\begin{aligned}
\mathrm{excess}_\gamma(\widehat{h}_M) &\leqslant 2\kappa + \frac{72\beta_M}{\tau_{M-1}\gamma} \cdot \theta^{\mathrm{val}}_{\bar{f}}\left(\mathcal{F}, \gamma/2, \sqrt{\beta_M/\tau_{M-1}}\right) \\
&\leqslant 2\kappa + O\left(\frac{\varepsilon^2}{\gamma} + \frac{\mathrm{Pdim}(\mathcal{F}) \cdot \log(T/\delta)}{T\gamma}\right) \cdot \theta^{\mathrm{val}}_{\bar{f}}\left(\mathcal{F}, \gamma/2, \sqrt{C_\delta/T}\right) \\
&= O\left(\varepsilon \cdot \bar{\theta} \cdot \log\left(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}\right)\right),
\end{aligned}$$

where we take $\bar{\theta} := \sup_{\iota>0} \theta_{\bar{f}}^{\mathrm{val}}(\mathcal{F}, \gamma/2, \iota)$ as an upper bound of $\theta_{\bar{f}}^{\mathrm{val}}(\mathcal{F}, \gamma/2, \sqrt{C_\delta/T})$, and use the fact that $T = \frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}$ and the assumptions that $\kappa \leqslant \varepsilon < \gamma$.

We now analyze the label complexity (note that the sampling process of Algorithm 3 stops at time $t = \tau_{M-1}$). Note that $\mathbb{E}[\mathbb{1}(Q_t = 1) \mid \mathfrak{F}_{t-1}] = \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x) = 1)]$ for any epoch $m \geqslant 2$ and time step t within epoch m. Combine Lemma 2.30 with Lemma 2.54 leads to

$$
\begin{aligned}
&\sum_{t=1}^{\tau_{M-1}} \mathbb{1}(Q_t = 1) \\
&\leqslant \frac{3}{2} \sum_{t=1}^{\tau_{M-1}} \mathbb{E}[\mathbb{1}(Q_t = 1) \mid \mathfrak{F}_{t-1}] + 4\log\delta^{-1} \\
&\leqslant 3 + \frac{3}{2} \sum_{m=2}^{M-1} \frac{(\tau_m - \tau_{m-1}) \cdot 36\beta_m}{\tau_{m-1}\gamma^2} \cdot \theta_{\bar{f}}^{\mathrm{val}}\left(\mathcal{F}, \gamma/2, \sqrt{\beta_m/\tau_{m-1}}\right) + 4\log\delta^{-1} \\
&\leqslant 3 + 48 \sum_{m=2}^{M-1} \frac{\beta_m}{\gamma^2} \cdot \theta_{\bar{f}}^{\mathrm{val}}\left(\mathcal{F}, \gamma/2, \sqrt{\beta_m/\tau_{m-1}}\right) + 4\log\delta^{-1} \\
&\leqslant 3 + 4\log\delta^{-1} + O\left(\frac{M^2\cdot\varepsilon^2\cdot T}{\gamma^2} + \frac{M^2\cdot C_\delta}{\gamma^2}\right) \cdot \theta_{\bar{f}}^{\mathrm{val}}\left(\mathcal{F}, \gamma/2, \sqrt{C_\delta/T}\right) \\
&= O\left(\frac{\bar{\theta}\,\mathrm{Pdim}(\mathcal{F})}{\gamma^2} \cdot \left(\log\left(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}\right)\right)^2 \cdot \log\left(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}\right)\right)
\end{aligned}
$$

with probability at least $1 - 2\delta$ (due to an additional application of Lemma 2.30); where we use the fact that $T = \frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}$ and the assumptions that $\kappa \leqslant \varepsilon < \gamma$ as before. □

Theorem 2.51. *Algorithm 3 can be efficiently implemented via the regression oracle and enjoys the same theoretical guarantees stated in Theorem 2.22. The number of oracle calls needed is $\widetilde{O}(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma^3})$ for a general set of regression functions $\mathcal{F}$, and $\widetilde{O}(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma})$ when $\mathcal{F}$ is convex and closed under pointwise convergence. The per-example inference time of the learned $\widehat{h}_M$ is $\widetilde{O}(\frac{1}{\gamma^2}\log^2(\frac{\mathrm{Pdim}(\mathcal{F})}{\varepsilon}))$ for general $\mathcal{F}$, and $\widetilde{O}(\log\frac{1}{\gamma})$ when $\mathcal{F}$ is convex and closed under pointwise convergence.*

Proof. Note that classifier $\widehat{h}_m$ and query function q_m in Algorithm 3 are constructed in the way as the ones in Algorithm 1, Thus, Algorithm 3 can be efficiently implemented in the same way as discussed in Theorem 2.8, and enjoys the same per-round computational complexities. The total computational complexity is then achieved by multiplying the per-round computational complexity by $T = \frac{\text{Pdim}(\mathcal{F})}{\varepsilon\gamma}$. □

2.5.6.2 Supporting Lemmas

We use the same notations defined in Section 2.5.3, except $\widehat{h}_m$, g_m and β_m are defined differently. We adapt the proofs Theorem 2.7 (in Section 2.5.3) to deal with model misspecification.

Note that although we do not have $f^\star \in \mathcal{F}$ anymore, one can still define random variables of the form $M_t(f)$, and guarantees in Lemma 2.33 still hold. We use $\mathcal{E}$ to denote the good event considered in Lemma 2.33, we analyze under this event through out the rest of this section. We also only analyze under the assumption of Theorem 2.22, i.e., $\kappa^2 \leqslant \varepsilon$.

Lemma 2.52. *Fix any epoch $m \in [M]$. We have*

$$\widehat{R}_m(\bar{f}) \leqslant \widehat{R}_m(f^\star) + \frac{3}{2}\cdot\kappa^2\tau_{m-1} + C_\delta,$$

where $C_\delta := 8\log\left(\frac{|\mathcal{F}|\cdot T^2}{\delta}\right)$.

Proof. From Lemma 2.33 we know that

$$\begin{aligned}\widehat{R}_m(\bar{f}) - \widehat{R}_m(f^\star) &\leqslant \sum_{t=1}^{\tau_{m-1}} \frac{3}{2}\cdot\mathbb{E}_t\left[Q_t\big(\bar{f}(x_t) - f^\star(x_t)\big)^2\right] + C_\delta \\ &\leqslant \frac{3}{2}\cdot\kappa^2\tau_{m-1} + C_\delta,\end{aligned}$$

where we use the fact that $\mathbb{E}_t[y_t \mid x_t] = f^\star(x_t)$ (and thus $\mathbb{E}_t[M_t(\bar{f})] = \mathbb{E}_t[Q_t(\bar{f}(x_t) - f^\star(x_t))^2]$) on the first line; and use the fact $\sup_x|\bar{f}(x) - f^\star(x)| \leqslant \kappa$ on the second line. □

Lemma 2.53. *The followings hold true:*

1. $\bar{f} \in \mathcal{F}_m$ *for any* $m \in [M]$.
2. $\sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t[M_t(f)] \leqslant 4\beta_m$ *for any* $f \in \mathcal{F}_m$.
3. $\sum_{t=1}^{\tau_{m-1}} \mathbb{E}[Q_t(x_t)(f(x_t) - \bar{f}(x_t))^2] \leqslant 9\beta_m$ *for any* $f \in \mathcal{F}_m$.
4. $\mathcal{F}_{m+1} \subseteq \mathcal{F}_m$ *for any* $m \in [M-1]$.

Proof. 1. Fix any epoch $m \in [M]$. By Lemma 2.33, we have $\widehat{R}_m(f^\star) \leqslant \widehat{R}_m(f) + C_\delta/2$ for any $f \in \mathcal{F}$. Combining this with Lemma 2.52 leads to

$$\begin{aligned}\widehat{R}_m(\bar{f}) &\leqslant \widehat{R}_m(f) + \frac{3}{2} \cdot \left(\kappa^2 \tau_{m-1} + C_\delta\right) \\ &\leqslant \widehat{R}_m(f) + \beta_m,\end{aligned}$$

for any $f \in \mathcal{F}$, where the second line comes from the definition of β_m (recall that we have $\kappa \leqslant \varepsilon$ by assumption). We thus have $\bar{f} \in \mathcal{F}_m$ for any $m \in [M]$.

2. Fix any $f \in \mathcal{F}_m$. With Lemma 2.33, we have

$$\begin{aligned}\sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t[M_t(f)] &\leqslant 2 \sum_{t=1}^{\tau_{m-1}} M_t(f) + C_\delta \\ &= 2\widehat{R}_m(f) - 2\widehat{R}_m(f^\star) + C_\delta \\ &\leqslant 2\widehat{R}_m(f) - 2\widehat{R}_m(\bar{f}) + 3\kappa^2 \tau_{m-1} + 3C_\delta \\ &\leqslant 2\widehat{R}_m(f) - 2\widehat{R}_m(\widehat{f}_m) + 3\kappa^2 \tau_{m-1} + 3C_\delta \\ &\leqslant 2\beta_m + 3\kappa^2 \tau_{m-1} + 3C_\delta \\ &\leqslant 4\beta_m,\end{aligned}$$

where the third line comes from Lemma 2.52; the fourth line comes from the fact that $\widehat{f}_m$ is the minimizer of $\widehat{R}_m(\cdot)$; and the fifth line comes from the fact that $f \in \mathcal{F}_m$.

3. Fix any $f \in \mathcal{F}_m$. With Lemma 2.33, we have

$$\begin{aligned}
&\sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t[Q_t(x_t)(f(x_t) - \bar{f}(x_t))^2] \\
&= \sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t[Q_t(x_t)((f(x_t) - f^\star(x_t)) + (f^\star(x_t) - \bar{f}(x_t)))^2] \\
&\leqslant 2 \sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t[Q_t(x_t)(f(x_t) - f^\star(x_t))^2] + 2\tau_{m-1}\kappa^2 \\
&= 2 \sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t[M_t(f)] + 2\tau_{m-1}\kappa^2 \\
&\leqslant 8\beta_m + 2\tau_{m-1}\kappa^2 \\
&\leqslant 9\beta_m,
\end{aligned}$$

where we use $(a+b)^2 \leqslant a^2 + b^2$ on the second line; and use statement 2 on the fourth line.

4. Fix any $f \in \mathcal{F}_{m+1}$. We have

$$\begin{aligned}
&\widehat{R}_m(f) - \widehat{R}_m(\widehat{f}_m) \\
&\leqslant \widehat{R}_m(f) - \widehat{R}_m(f^\star) + \frac{C_\delta}{2} \\
&= \widehat{R}_{m+1}(f) - \widehat{R}_{m+1}(f^\star) - \sum_{t=\tau_{m-1}+1}^{\tau_m} M_t(f) + \frac{C_\delta}{2} \\
&\leqslant \widehat{R}_{m+1}(f) - \widehat{R}_{m+1}(\bar{f}) + \frac{3}{2}\kappa^2\tau_m + C_\delta - \sum_{t=\tau_{m-1}+1}^{\tau_m} \mathbb{E}_t[M_t(f)]/2 + C_\delta \\
&\leqslant \widehat{R}_{m+1}(f) - \widehat{R}_{m+1}(\widehat{f}_{m+1}) + \frac{3}{2}\kappa^2\tau_m + 2C_\delta \\
&\leqslant \beta_{m+1} + \frac{3}{2}\kappa^2\tau_m + 2C_\delta \\
&\leqslant \beta_m,
\end{aligned}$$

where the first line comes from Lemma 2.33; the third line comes from Lemma 2.52 and Lemma 2.33; the fourth line comes from the fact that $\widehat{f}_{m+1}$ is the minimizer with respect to $\widehat{R}_{m+1}$ and Lemma 2.33; the last line comes from the definition of β_m.

□

Since the classifier $\widehat{h}_m$ and query function g_m are defined in the same way as in Algorithm 1, Lemma 2.36 holds true for Algorithm 3 as well. As a result of that, Lemma 2.37 and Lemma 2.38 hold true with minor modifications. We present the modified versions below, whose proofs follow similar steps as in Lemma 2.37 and Lemma 2.38 but replace $f^\star$ with $\widehat{f}$ (and thus using concentration results derived in Lemma 2.53).

Lemma 2.54. *Fix any epoch $m \geqslant 2$. We have*

$$\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x)=1)] \leqslant \frac{36\beta_m}{\tau_{m-1}\gamma^2}\cdot\theta_{\bar{f}}^{\mathrm{val}}\Big(\mathcal{F},\gamma/2,\sqrt{\beta_m/\tau_{m-1}}\Big).$$

Lemma 2.55. *Fix any epoch $m \geqslant 2$. We have*

$$\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x)=1)\cdot w(x;\mathcal{F}_m)] \leqslant \frac{36\beta_m}{\tau_{m-1}\gamma}\cdot\theta_{\bar{f}}^{\mathrm{val}}\Big(\mathcal{F},\gamma/2,\sqrt{\beta_m/\tau_{m-1}}\Big).$$

Lemma 2.56. *Fix any $m \in [M]$. We have* $\mathsf{excess}_\gamma(\widehat{h}_m;x) \leqslant 2\kappa$ *if* $g_m(x)=0$.

Proof. Recall that

$$\begin{aligned}\mathsf{excess}_\gamma(\widehat{h};x) = {}& \mathbb{1}\big(\widehat{h}(x)\neq\perp\big)\cdot\big(\mathbb{P}_{y|x}\big(y\neq\mathrm{sign}(\widehat{h}(x))\big)-\mathbb{P}_{y|x}\big(y\neq\mathrm{sign}(h^\star(x))\big)\big)\\ &+\mathbb{1}\big(\widehat{h}(x)=\perp\big)\cdot\big((1/2-\gamma)-\mathbb{P}_{y|x}\big(y\neq\mathrm{sign}(h^\star(x))\big)\big).\end{aligned}$$

We now analyze the event $\{g_m(x)=0\}$ in two cases.

Case 1: $\widehat{h}_m(x)=\perp$**.**

Since $\bar{f}(x) \in [\mathsf{lcb}(x;\mathcal{F}_m),\mathsf{ucb}(x;\mathcal{F}_m)]$ by Lemma 2.53, we know that $\eta(x) = f^\star(x) \in [\frac{1}{2}-\gamma-\kappa,\frac{1}{2}+\gamma+\kappa]$ and thus $\mathbb{P}_y\big(y\neq\mathrm{sign}(h^\star(x))\big)\geqslant\frac{1}{2}-\gamma-\kappa$. As a result, we have $\mathsf{excess}_\gamma(\widehat{h}_m;x)\leqslant\kappa$.

Case 2: $\widehat{h}_m(x) \neq \perp$ **but** $\frac{1}{2} \notin (\mathsf{lcb}(x;\mathcal{F}_m), \mathsf{ucb}(x;\mathcal{F}_m))$**.**

We clearly have $\mathsf{excess}_\gamma(\widehat{h}_m; x) \leqslant 0$ if $\mathrm{sign}(\widehat{h}_m(x)) = \mathrm{sign}(h^\star(x))$. Now consider the case when $\mathrm{sign}(\widehat{h}_m(x)) \neq \mathrm{sign}(h^\star(x))$. Since $\bar{f}(x) \in [\mathsf{lcb}(x;\mathcal{F}_m), \mathsf{ucb}(x;\mathcal{F}_m)]$ and $|\bar{f}(x) - f^\star(x)| \leqslant \kappa$, we must have $|f^\star(x) - 1/2| \leqslant \kappa$ in that case, which leads to $\mathsf{excess}_\gamma(\widehat{h}_m; x) \leqslant 2|f^\star(x) - 1/2| \leqslant 2\kappa$. □

3 ACTIVE LEARNING WITH NEURAL NETWORKS

Deep neural networks have great representation power, but typically require large numbers of training examples. This motivates deep active learning methods that can significantly reduce the amount of labeled training data. Empirical successes of deep active learning have been recently reported in the literature, however, rigorous label complexity guarantees of deep active learning have remained elusive. This constitutes a significant gap between theory and practice. This chapter tackles this gap by providing the first near-optimal label complexity guarantees for deep active learning. The key insight is to study deep active learning from the nonparametric classification perspective. Under standard low noise conditions, we show that active learning with neural networks can provably achieve the minimax label complexity, up to disagreement coefficient and other logarithmic terms. When equipped with an abstention option, we further develop an efficient deep active learning algorithm that achieves $\mathsf{polylog}(\frac{1}{\varepsilon})$ label complexity, without any low noise assumptions. We also provide extensions of our results beyond the commonly studied Sobolev/Hölder spaces and develop label complexity guarantees for learning in Radon BV^2 spaces, which have recently been proposed as natural function spaces associated with neural networks.

3.1 Introduction

We study active learning with neural network hypothesis classes, sometimes known as *deep active learning*. Active learning agent proceeds by selecting the most informative data points to label: The goal of active learning is to achieve the same accuracy achievable by passive learning, but with much fewer label queries (Settles, 2009; Hanneke, 2014). When the hypothesis class is a set of neural networks, the learner further benefits from the representation power of deep neural networks, which has driven the successes of passive learning in the past decade (Krizhevsky et al., 2012; LeCun et al., 2015). With these added benefits, deep active learning has

become a popular research area, with empirical successes observed in many recent papers. (Sener and Savarese, 2018; Ash et al., 2019; Citovsky et al., 2021; Ash et al., 2021; Kothawade et al., 2021; Emam et al., 2021; Ren et al., 2021). However, due to the difficulty of analyzing a set of neural networks, rigorous label complexity guarantees for deep active learning have remained largely elusive.

To the best of our knowledge, there are only two papers (Karzand and Nowak, 2020; Wang et al., 2021) that have made the attempts at theoretically quantifying active learning gains with neural networks. While insightful views are provided, these two works have their own limitations. The guarantees provided in Karzand and Nowak (2020) only work in the 1d case where data points are uniformly sampled from $[0, 1]$ and labeled by a well-seperated piece-wise constant function in a noise-free way (i.e., without any labeling noise). Wang et al. (2021) study deep active learning by linearizing the neural network at its random initialization and then analyzing it as a linear function; moreover, as the authors agree, their error bounds and label complexity guarantees can in fact be *vacuous* in certain cases. Thus, it's fair to say that up to now researchers have not identified cases where deep active learning are provably near minimax optimal (or even with provably non-vacuous guarantees), which constitutes a significant gap between theory and practice.

In this chapter, we bridge this gap by providing the first near-optimal label complexity guarantees for deep active learning. We obtain insights from the non-parametric setting where the conditional probability (of taking a positive label) is assumed to be a smooth function (Tsybakov, 2004; Audibert and Tsybakov, 2007). Previous nonparametric active learning algorithms proceed by partitioning the action space into exponentially many sub-regions (e.g., partitioning the unit cube $[0, 1]^d$ into ε^{-d} sub-cubes each with volume ε^d), and then conducting local mean (or some higher-order statistics) estimation within each sub-region (Castro and Nowak, 2008; Minsker, 2012; Locatelli et al., 2017, 2018; Shekhar et al., 2021; Kpotufe et al., 2021). We show that, with an appropriately chosen set of neural networks that *globally* approximates the smooth regression function, one can in fact recover the minimax label complexity for active learning, up to disagreement coefficient

(Hanneke, 2007, 2014) and other logarithmic factors. Our results are established by (i) identifying the "right tools" to study neural networks (ranging from approximation results (Yarotsky, 2017, 2018) to complexity measure of neural networks (Bartlett et al., 2019)), and (ii) developing novel extensions of agnostic active learning algorithms (Balcan et al., 2006; Hanneke, 2007, 2014) to work with a set of neural networks.

While matching the minimax label complexity in nonparametric active learning is existing, such minimax results scale as $\Theta(\text{poly}(\frac{1}{\varepsilon}))$ (Castro and Nowak, 2008; Locatelli et al., 2017) and do not resemble what is practically observed in deep active learning: A fairly accurate neural network classifier can be obtained by training with only a few labeled data points. Inspired by recent results in *parametric* active learning with abstention (Puchkin and Zhivotovskiy, 2021; Zhu and Nowak, 2022b), we develop an oracle-efficient algorithm showing that deep active learning provably achieves $\text{polylog}(\frac{1}{\varepsilon})$ label complexity when equipped with an abstention option (Chow, 1970). Our algorithm not only achieves an exponential saving in label complexity (*without any low noise assumptions*), but is also highly practical: In real-world scenarios such as medical imaging, it makes more sense for the classifier to abstain from making prediction on hard examples (e.g., those that are close to the boundary), and ask medical experts to make the judgments.

3.1.0.1 Proper Abstention and computational efficiency

3.1.1 Problem Setting

Let $\mathcal{X}$ denote the instance space and $\mathcal{Y}$ denote the label space. We focus on the binary classification problem where $\mathcal{Y} := \{+1, -1\}$. The joint distribution over $\mathcal{X} \times \mathcal{Y}$ is denoted as $\mathcal{D}_{\mathcal{X}\mathcal{Y}}$. We use $\mathcal{D}_{\mathcal{X}}$ to denote the marginal distribution over the instance space $\mathcal{X}$, and use $\mathcal{D}_{\mathcal{Y}|x}$ to denote the conditional distribution of $\mathcal{Y}$ with respect to any $x \in \mathcal{X}$. We consider the standard active learning setup where $x \sim \mathcal{D}_{\mathcal{X}}$ but its label $y \sim \mathcal{D}_{\mathcal{Y}|x}$ is only observed after issuing a label query. We define $\eta(x) := \mathbb{P}_{y \sim \mathcal{D}_{\mathcal{Y}|x}}(y = +1)$ as the conditional probability of taking a positive label. The Bayes optimal classifier $h^\star$ can thus be expressed as $h^\star(x) := \text{sign}(2\eta(x) - 1)$. For any classifier

$h : \mathcal{X} \to \mathcal{Y}$, its (standard) error is calculated as $\mathrm{err}(h) := \mathbb{P}_{(x,y)\sim\mathcal{D}_{\mathcal{X}\mathcal{Y}}}(h(x) \neq y)$; and its (standard) excess error is defined as $\mathrm{excess}(h) := \mathrm{err}(h) - \mathrm{err}(h^\star)$. Our goal is to learn an accurate classifier with a small number of label querying.

The nonparametric setting. We consider the nonparametric setting where the conditional probability η is characterized by a smooth function. Fix any $\alpha \in \mathbb{N}_+$, the *Sobolev norm* of a function $f : \mathcal{X} \to \mathbb{R}$ is defined as $\|f\|_{\mathcal{W}^{\alpha,\infty}} := \max_{\bar{\alpha},|\bar{\alpha}|\leqslant\alpha} \operatorname{ess\,sup}_{x\in\mathcal{X}} |D^{\alpha} f(x)|$, where $\alpha = (\alpha_1, \ldots, \alpha_d)$, $|\alpha| = \sum_{i=1}^{d} \alpha_i$ and $D^{\alpha} f$ denotes the standard α-th weak derivative of f. The unit ball in the Sobolev space is defined as $\mathcal{W}_1^{\alpha,\infty}(\mathcal{X}) := \{f : \|f\|_{\mathcal{W}^{\alpha,\infty}} \leqslant 1\}$. Following the convention of nonparametric active learning (Castro and Nowak, 2008; Minsker, 2012; Locatelli et al., 2017, 2018; Shekhar et al., 2021; Kpotufe et al., 2021), we assume $\mathcal{X} = [0,1]^d$ and $\eta \in \mathcal{W}_1^{\alpha,\infty}(\mathcal{X})$ (except in Section 3.4).

Neural Networks. We consider *feedforward neural networks* with Rectified Linear Unit (ReLU) activation function, which is defined as $\mathrm{ReLU}(x) := \max\{x, 0\}$. Each neural network $f_{\mathsf{dnn}} : \mathcal{X} \to \mathbb{R}$ consists of several input units (which corresponds to the covariates of $x \in \mathcal{X}$), one output unit (which corresponds to the prediction in $\mathbb{R}$), and multiple hidden computational units. Each hidden computational unit takes inputs $\{\bar{x}_i\}_{i=1}^{N}$ (which are outputs from previous layers) and perform the computation $\mathrm{ReLU}(\sum_{i=1}^{N} w_i \bar{x}_i + b)$ with *adjustable* parameters $\{w_i\}_{i=1}^{N}$ and b; the output unit performs the same operation, but without the ReLU nonlinearity. We use W to denote the total number of parameters of a neural network, and L to denote the depth of the neural network.

3.1.2 Contributions and Organization

Neural networks are known to be universal approximators (Cybenko, 1989; Hornik, 1991). In this chapter, we argue that, in both passive and active regimes, the universal approximatability makes neural networks "universal classifiers" for classification problems: With an appropriately chosen set of neural networks, one can recover

known minimax rates (up to disagreement coefficients in the active setting) in the rich nonparametric regimes.[1] We provide informal statements of our main results in the sequel, with detailed statements and associated definitions/algorithms deferred to later sections.

In Section 3.2, we analyze the label complexity of deep active learning under the standard Tsybakov noise condition with smoothness parameter $\beta \geqslant 0$ (Tsybakov, 2004). Let $\mathcal{H}_{\mathsf{dnn}}$ be an appropriately chosen set of neural network classifiers and denote $\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon)$ as the disagreement coefficient (Hanneke, 2007, 2014) at level ε. We develop the following label complexity guarantees for deep active learning.

Theorem 3.1 (Informal)**.** *There exists an algorithm that returns a neural network classifier* $\widehat{h} \in \mathcal{H}_{\mathsf{dnn}}$ *with excess error* $\widetilde{O}(\varepsilon)$ *after querying* $\widetilde{O}(\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) \cdot \varepsilon^{-\frac{d+2\alpha}{\alpha+\alpha\beta}})$ *labels.*

The label complexity presented in Theorem 3.1 matches the active learning lower bound $\Omega(\varepsilon^{-\frac{d+2\alpha}{\alpha+\alpha\beta}})$ (Locatelli et al., 2017) up to the dependence on the disagreement coefficient (and other logarithmic factors). Since $\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon) \leqslant \varepsilon^{-1}$ by definition, the label complexity presented in Theorem 3.1 is never worse than the passive learning rates $\widetilde{\Theta}(\varepsilon^{-\frac{d+2\alpha+\alpha\beta}{\alpha+\alpha\beta}})$ (Audibert and Tsybakov, 2007). We also discover conditions under which the disagreement coefficient with respect to a set of neural network classifiers can be properly bounded, i.e., $\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon) = o(\varepsilon^{-1})$ (implying strict improvement over passive learning) and $\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon) = o(1)$ (implying matching active learning lower bound).

In Section 3.3, we develop label complexity guarantees for deep active learning when an additional abstention option is allowed (Chow, 1970; Puchkin and Zhivotovskiy, 2021; Zhu and Nowak, 2022b). Suppose a cost (e.g. 0.49) that is marginally smaller than random guessing (which has expected cost 0.5) is incurred whenever the classifier abstains from making a predication, we develop the following label complexity guarantees for deep active learning.

[1] As a byproduct, our results also provide a new perspective on nonparametric active learning through the lens of neural network approximations. Nonparametric active learning was previously tackled through space partitioning and local estimations over exponentially many sub-regions (Castro and Nowak, 2008; Minsker, 2012; Locatelli et al., 2017, 2018; Shekhar et al., 2021; Kpotufe et al., 2021).

Theorem 3.2 (Informal)**.** *There exists an efficient algorithm that constructs a neural network classifier* $\widehat{\mathsf{h}}_{\mathsf{dnn}}$ *with Chow's excess error* $\widetilde{\mathrm{O}}(\varepsilon)$ *after querying* $\mathrm{polylog}(\frac{1}{\varepsilon})$ *labels.*

The above $\mathrm{polylog}(\frac{1}{\varepsilon})$ label complexity bound is achieved *without any low noise assumptions*. Such exponential label savings theoretically justify the great empirical performances of deep active learning observed in practice (e.g., in Sener and Savarese (2018)): It suffices to label a few data points to achieve a high accuracy level. Moreover, apart from an initialization step, our algorithm (Algorithm 7) developed for Theorem 3.2 can be *efficiently* implemented in $\widetilde{\mathrm{O}}(\varepsilon^{-1})$ time, given a convex loss regression oracle over an appropriately chosen set of neural networks; in practice, the regression oracle can be approximated by running stochastic gradient descent.

Technical contributions. Besides identifying the "right tools" (ranging from approximation results (Yarotsky, 2017, 2018) to complexity analyses (Bartlett et al., 2019)) to analyze deep active learning, our theoretical guarantees are empowered by novel extensions of active learning algorithms *under neural network approximations*. In particular, we deal with approximation error in active learning under Tsybakov noise, and identify conditions that greatly relax the approximation requirement in the learning with abstention setup; we also analyze the disagreement coefficient, both classifier-based and value function-based, with a set of neural networks.These analyses together lead to our main results for deep active learning (e.g., Theorem 3.1 and Theorem 3.2). More generally, we establish a bridge between approximation theory and active learning; we provide these general guarantees in Section 3.6 (under Tsybakov noise) and Section 3.7 (with the abstention option), which can be of independent interests. Benefited from these generic algorithms and guarantees, in Section 3.4, we extend our results into learning smooth functions in the Radon BV^2 space (Ongie et al., 2020; Parhi and Nowak, 2021, 2022b,a; Unser, 2022), which is recently proposed as a natural space to analyze neural networks.

3.1.3 Additional Related Work

Active learning concerns about learning accurate classifiers without extensive human labeling. One of the earliest work of active learning dates back to the CAL algorithm proposed by Cohn et al. (1994), which set the cornerstone for *disagreement-based* active learning. Since then, a long line of work have been developed, either directly working with a set classifier (Balcan et al., 2006; Hanneke, 2007; Dasgupta et al., 2007; Beygelzimer et al., 2009, 2010; Huang et al., 2015; Cortes et al., 2019) or work with a set of regression functions (Krishnamurthy et al., 2017, 2019). These work mainly focus on the parametric regime (e.g., learning with a set of linear classifiers), and their label complexities rely on the boundedness of the so-called disagreement coefficient (Hanneke, 2007, 2014; Friedman, 2009). Active learning in the nonparametric regime has been analyzed in Castro and Nowak (2008); Minsker (2012); Locatelli et al. (2017, 2018); Kpotufe et al. (2021). These algorithms rely on partitioning of the input space $\mathcal{X} \subseteq [0,1]^d$ into exponentially (in dimension) many small cubes, and then conduct local mean (or some higher-order statistics) estimation within each small cube.

It is well known that, in the worst case, active learning exhibits no label complexity gains over the passive counterpart (Kääriäinen, 2006). To bypass these worst-case scenarios, active learning has been popularly analyzed under the so-called Tsybakov low noise conditions (Tsybakov, 2004). Under Tsybakov noise conditions, active learning has been shown to be strictly superior than passive learning in terms of label complexity (Castro and Nowak, 2008; Locatelli et al., 2017). Besides analyzing active learning under favorable low noise assumptions, more recently, researchers consider active learning with an abstention option and analyze its label complexity under Chow's error (Chow, 1970). In particular, Puchkin and Zhivotovskiy (2021); Zhu and Nowak (2022b) develop active learning algorithms with $\mathrm{polylog}(\frac{1}{\varepsilon})$ label complexity when analyzed under Chow's excess error. Shekhar et al. (2021) study nonparametric active learning under a different notion of the Chow's excess error, and propose algorithms with $\mathrm{poly}(\frac{1}{\varepsilon})$ label complexity; their algorithms follow similar procedures of those partition-based nonparametric

active learning algorithms (e.g., Minsker (2012); Locatelli et al. (2017)).

Inspired by the success of deep learning in the passive regime, active learning with neural networks has been extensively explored in recent years (Sener and Savarese, 2018; Ash et al., 2019; Citovsky et al., 2021; Ash et al., 2021; Kothawade et al., 2021; Emam et al., 2021; Ren et al., 2021). Great empirical performances are observed in these papers, however, rigorous label complexity guarantees have largely remains elusive (except in Karzand and Nowak (2020); Wang et al. (2021), with limitations discussed before). We bridge the gap between practice and theory by providing the first near-optimal label complexity guarantees for deep active learning. Our results are built upon approximation results of deep neural networks (Yarotsky, 2017, 2018; Parhi and Nowak, 2022a) and VC/pseudo dimension analyses of neural networks with given structures (Bartlett et al., 2019).

3.2 Label Complexity of Deep Active Learning

We analyze the label complexity of deep active learning in this section. We first introduce the Tsybakov noise condition in Section 3.2.1, and then identify the "right tools" to analyze classification problems with neural network classifiers in Section 3.2.2 (where we also provide passive learning guarantees). We establish our main active learning guarantees in Section 3.2.3.

3.2.1 Tsybakov Noise Condition

It is well known that active learning exhibits no label complexity gains over the passive counterpart without additional low noise assumptions (Kääriäinen, 2006). We next introduce the Tsybokov low noise condition (Tsybakov, 2004), which has been extensively analyzed in active learning literature.

Definition 3.3 (Tsybakov noise)**.** *A distribution $\mathcal{D}_{XY}$ satisfies the Tsybakov noise condition with parameter $\beta \geqslant 0$ and a universal constant $c \geqslant 1$ if, $\forall \tau > 0$,*

$$\mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \tau) \leqslant c\,\tau^{\beta}.$$

The case with $\beta = 0$ corresponds to the general case *without* any low noise conditions, where no active learning algorithm can outperform the passive counterpart (Audibert and Tsybakov, 2007; Locatelli et al., 2017). We use $\mathcal{P}(\alpha, \beta)$ to denote the set of distributions satisfying: (i) the smoothness conditions introduced in Section 3.1.1 with parameter $\alpha > 0$; and (ii) the Tsybakov low noise condition (i.e., Definition 3.3) with parameter $\beta \geqslant 0$. We assume $\mathcal{D}_{XY} \in \mathcal{P}(\alpha, \beta)$ in the rest of Section 3.2. As in Castro and Nowak (2008); Hanneke (2014), we assume the knowledge of noise/smoothness parameters.

3.2.2 Approximation and Expressiveness of Neural Networks

Neural networks are known to be universal approximators (Cybenko, 1989; Hornik, 1991): For any continuous function $g : \mathcal{X} \to \mathbb{R}$ and any error tolerance $\kappa > 0$, there exists a large enough neural network f_{dnn} such that $\|f_{dnn} - g\|_\infty := \sup_{x \in \mathcal{X}} |f_{dnn}(x) - g(x)| \leqslant \kappa$. Recently, *non-asympototic* approximation rates by ReLU neural networks have been developed for smooth functions in the Sobolev space, which we restate in the following.[2]

Theorem 3.4 (Yarotsky (2017))**.** *Fix any $\kappa > 0$. For any $f^\star = \eta \in \mathcal{W}_1^{\alpha,\infty}([0,1]^d)$, there exists a neural network f_{dnn} with $W = O(\kappa^{-\frac{d}{\alpha}} \log \frac{1}{\kappa})$ total number of parameters arranged in $L = O(\log \frac{1}{\kappa})$ layers such that $\|f_{dnn} - f^\star\|_\infty \leqslant \kappa$.*

The architecture of the neural network f_{dnn} appearing in the above theorem only depends on the smooth function space $\mathcal{W}_1^{\alpha,\infty}([0,1]^d)$, but otherwise is independent of the true regression function $f^\star$; also see Yarotsky (2017) for details. Let $\mathcal{F}_{dnn}$ denote the set of neural network *regression functions* with the same architecture. We construct a set of neural network *classifiers* by thresholding the regression function at $\frac{1}{2}$, i.e., $\mathcal{H}_{dnn} := \{h_f := \text{sign}(2f(x) - 1) : f \in \mathcal{F}_{dnn}\}$. The next result concerns about the expressiveness of the neural network classifiers, in terms of a well-known complexity measure: the VC dimension (Vapnik and Chervonenkis, 1971).

[2] As in Yarotsky (2017), we hide constants that are potentially α-dependent and d-dependent into the Big-Oh notation.

Theorem 3.5 (Bartlett et al. (2019)). *Let $\mathcal{H}_{\mathsf{dnn}}$ be a set of neural network classifiers of the same architecture and with W parameters arranged in L layers. We then have*

$$\Omega(WL\log(W/L)) \leqslant \mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}}) \leqslant O(WL\log(W)).$$

With these tools, we can construct a set of neural network classifiers $\mathcal{H}_{\mathsf{dnn}}$ such that (i) the best in-class classifier $\check{h} \in \mathcal{H}_{\mathsf{dnn}}$ has small excess error, and (ii) $\mathcal{H}_{\mathsf{dnn}}$ has a well-controlled VC dimension that is proportional to smooth/noise parameters. More specifically, we have the following proposition.

Proposition 3.6. *Suppose $\mathcal{D}_{\mathcal{X}\mathcal{Y}} \in \mathcal{P}(\alpha, \beta)$. One can construct a set of neural network classifier $\mathcal{H}_{\mathsf{dnn}}$ such that the following two properties hold simultaneously:*

$$\inf_{h \in \mathcal{H}_{\mathsf{dnn}}} \mathrm{err}(h) - \mathrm{err}(h^\star) = O(\varepsilon) \quad \textit{and} \quad \mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}}) = \widetilde{O}(\varepsilon^{-\frac{d}{\alpha(1+\beta)}}).$$

With the approximation results obtained above, to learn a classifier with $O(\varepsilon)$ excess error, one only needs to focus on a set of neural networks $\mathcal{H}_{\mathsf{dnn}}$ with a well-controlled VC dimension. As a warm-up, we first analyze the label complexity of such procedure in the passive regime (with fast rates).

Theorem 3.7. *Suppose $\mathcal{D}_{\mathcal{X}\mathcal{Y}} \in \mathcal{P}(\alpha, \beta)$. Fix any $\varepsilon, \delta > 0$. Let $\mathcal{H}_{\mathsf{dnn}}$ be the set of neural network classifiers constructed in Proposition 3.6. With $n = \widetilde{O}(\varepsilon^{-\frac{d+2\alpha+\alpha\beta}{\alpha(1+\beta)}})$ i.i.d. sampled points, with probability at least $1-\delta$, the empirical risk minimizer $\widehat{h} \in \mathcal{H}_{\mathsf{dnn}}$ achieves excess error $O(\varepsilon)$.*

The label complexity results obtained in Theorem 3.7 matches, up to logarithmic factors, the passive learning lower bound $\Omega(\varepsilon^{-\frac{d+2\alpha+\alpha\beta}{\alpha(1+\beta)}})$ established in Audibert and Tsybakov (2007), indicating our proposed learning procedure *with a set of neural networks* is near minimax optimal.

3.2.3 Deep Active Learning and Guarantees

The passive learning procedure presented in the previous section treats every data point equally, i.e., it requests the label of every data point. Active learning

reduces the label complexity by only querying labels of data points that are "more important". We present deep active learning results in this section. Our algorithm (Algorithm 4) is inspired by RobustCAL (Balcan et al., 2006; Hanneke, 2007, 2014) and the seminal CAL algorithm (Cohn et al., 1994); we call our algorithm NeuralCAL to emphasize that it works with a set of neural networks.

For any accuracy level $\varepsilon > 0$, NeuralCAL first initialize a set of neural network classifiers $\mathcal{H}_0 := \mathcal{H}_{\mathsf{dnn}}$ such that (i) the best in-class classifier $\check{h} := \arg\min_{h \in \mathcal{H}_{\mathsf{dnn}}} \mathrm{err}(h)$ has excess error at most $O(\varepsilon)$, and (ii) the VC dimension of $\mathcal{H}_{\mathsf{dnn}}$ is upper bounded by $\widetilde{O}(\varepsilon^{-\frac{d}{\alpha(1+\beta)}})$ (see Section 3.2.2 for more details). NeuralCAL then runs in epochs of geometrically increasing lengths. At the beginning of epoch m, based on previously *labeled* data points, NeuralCAL updates a set of active classifier $\mathcal{H}_m$ such that, with high probability, the best classifier $\check{h}$ remains *uneliminated*. Within each epoch m, NeuralCAL only queries the label y of a data point x if it lies in the *region of disagreement* with respect to the current active set of classifier $\mathcal{H}_m$, i.e., $\mathsf{DIS}(\mathcal{H}_m) := \{x \in \mathcal{X} : \exists h_1, h_2 \in \mathcal{H}_m \text{ s.t. } h_1(x) \neq h_2(x)\}$. NeuralCAL returns any classifier $\widehat{h} \in \mathcal{H}_m$ that remains uneliminated after $M - 1$ epoch.

Algorithm 4 NeuralCAL

Input: Accuracy level $\varepsilon \in (0,1)$, confidence level $\delta \in (0,1)$.

Let $\mathcal{H}_{\mathsf{dnn}}$ be a set of neural networks classifiers constructed in Proposition 3.6.

Define $T := \varepsilon^{-\frac{2+\beta}{1+\beta}} \cdot \mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}})$, $M := \lceil \log_2 T \rceil$, $\tau_m := 2^m$ for $m \geqslant 1$ and $\tau_0 := 0$.

Define $\rho_m := O\left(\left(\frac{\mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}}) \cdot \log(\tau_{m-1}) \cdot \log(M/\delta)}{\tau_{m-1}}\right)^{\frac{1+\beta}{2+\beta}}\right)$ for $m \geqslant 2$ and $\rho_1 := 1$.

Define $\widehat{R}_m(h) := \sum_{t=1}^{\tau_{m-1}} Q_t \mathbb{1}(h(x_t) \neq y_t)$ with the convention that $\sum_{t=1}^{0} \ldots = 0$.

Initialize $\mathcal{H}_0 := \mathcal{H}_{\mathsf{dnn}}$.

for epoch $m = 1, 2, \ldots, M$ **do**

  Update active set $\mathcal{H}_m := \left\{h \in \mathcal{H}_{m-1} : \widehat{R}_m(h) \leqslant \inf_{h \in \mathcal{H}_{m-1}} \widehat{R}_m(h) + \tau_{m-1} \cdot \rho_m\right\}$

  if epoch $m = M$ **then**

    Return any classifier $\widehat{h} \in \mathcal{H}_M$.

  for time $t = \tau_{m-1} + 1, \ldots, \tau_m$ **do**

    Observe $x_t \sim \mathcal{D}_X$. Set $Q_t := \mathbb{1}(x_t \in \mathsf{DIS}(\mathcal{H}_m))$.

    if $Q_t = 1$ **then**

      Query the label y_t of x_t.

Since NeuralCAL only queries labels of data points lying in the region of disagreement, its label complexity should intuitively be related to how fast the region of disagreement shrinks. More formally, the rate of collapse of the (probability measure of) region of disagreement is captured by the (*classifier-based*) *disagreement coefficient* (Hanneke, 2007, 2014), which we introduce next.

Definition 3.8 (Classifier-based disagreement coefficient)**.** *For any ε_0 and classifier $h \in \mathcal{H}$, the classifier-based disagreement coefficient of h is defined as*

$$\theta_{\mathcal{H},h}(\varepsilon_0) := \sup_{\varepsilon > \varepsilon_0} \frac{\mathbb{P}_{x \sim \mathcal{D}_X}(\mathsf{DIS}(\mathcal{B}_{\mathcal{H}}(h, \varepsilon)))}{\varepsilon} \vee 1,$$

where $\mathcal{B}_{\mathcal{H}}(h, \varepsilon) := \{g \in \mathcal{H} : \mathbb{P}(x \in \mathcal{X} : g(x) \neq h(x)) \leqslant \varepsilon\}$. We also define $\theta_{\mathcal{H}}(\varepsilon_0) := \sup_{h \in \mathcal{H}} \theta_{\mathcal{H},h}(\varepsilon_0)$.

The guarantees of NeuralCAL follows from a more general analysis of RobustCAL under approximation. In particular, to achieve fast rates (under Tsybakov noise), previous analysis of RobustCAL requires that the Bayes classifier is in the class (or a Bernstein condition for every $h \in \mathcal{H}$) (Hanneke, 2014). These requirements are stronger compared to what we have in the case with neural network approximations. Our analysis extends the understanding of RobustCAL under approximation. We defer such general analysis to Section 3.6, and present the following guarantees.

Theorem 3.9. *Suppose $\mathcal{D}_{XY} \in \mathcal{P}(\alpha, \beta)$. Fix any $\varepsilon, \delta > 0$. With probability at least $1 - \delta$, Algorithm 4 returns a classifier $\widehat{h} \in \mathcal{H}_{\mathsf{dnn}}$ with excess error $\widetilde{O}(\varepsilon)$ after querying $\widetilde{O}(\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) \cdot \varepsilon^{-\frac{d+2\alpha}{\alpha+\alpha\beta}})$ labels.*

We next discuss in detail the label complexity of deep active learning proved in Theorem 3.9.

- Ignoring the dependence on disagreement coefficient, the label complexity appearing in Theorem 3.9 matches, up to logarithmic factors, the lower bound $\Omega(\varepsilon^{-\frac{d+2\alpha}{\alpha+\alpha\beta}})$ for active learning (Locatelli et al., 2017). At the same time, the label complexity appearing in Theorem 3.9 is *never worse* than the passive counterpart (i.e., $\widetilde{\Theta}(\varepsilon^{-\frac{d+2\alpha+\alpha\beta}{\alpha(1+\beta)}})$ since $\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) \leqslant \varepsilon^{-\frac{\beta}{1+\beta}}$.
- We also identify cases when $\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) = o(\varepsilon^{-\frac{\beta}{1+\beta}})$, indicating *strict* improvement over passive learning (e.g., when $\mathcal{D}_X$ is supported on countably many data points), and when $\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) = O(1)$, indicating matching the minimax active lower bound (e.g., when $\mathcal{D}_{XY}$ satisfies conditions such as *decomposibility* defined in Definition 3.34. See Section 3.8.2.2 for detailed discussion).[3]

Our algorithm and theorems lead to the following results, which could benefit both deep active learning and nonparametric learning communities.

[3]We remark that disagreement coefficient is usually bounded/analyzed under additional assumptions on $\mathcal{D}_{XY}$, even for simple cases with a set of linear classifiers (Friedman, 2009; Hanneke, 2014). The label complexity guarantees of partition-based nonparametric active algorithms (e.g., Castro and Nowak (2008)) do not depend on the disagreement coefficient, but they are analyzed under stronger assumptions, e.g., they require the strictly stronger membership querying oracle. See Wang (2011) for a discussion. We left a comprehensive analysis of the disagreement coefficient with a set of neural network classifiers for future work.

- **Near minimax optimal label complexity for deep active learning.** While empirical successes of deep active learning have been observed, rigorous label complexity analysis remains elusive except for two attempts made in Karzand and Nowak (2020); Wang et al. (2021). The guarantees provided in Karzand and Nowak (2020) only work in very special cases (i.e., data uniformly sampled from $[0, 1]$ and labeled by well-separated piece-constant functions in a noise-free way). Wang et al. (2021) study deep active learning in the NTK regime by linearizing the neural network at its random initialization and analyzing it as a linear function; moreover, as the authors agree, their error bounds and label complexity guarantees are *vacuous* in certain cases. On the other hand, our guarantees are minimax optimal, up to disagreement coefficient and other logarithmic factors, which bridge the gap between theory and practice in deep active learning.

- **New perspective on nonparametric learning.** Nonparametric learning of smooth functions have been mainly approached by partitioning-based methods (Tsybakov, 2004; Audibert and Tsybakov, 2007; Castro and Nowak, 2008; Minsker, 2012; Locatelli et al., 2017, 2018; Kpotufe et al., 2021) : Partition the unit cube $[0, 1]^d$ into exponentially (in dimension) many sub-cubes and conduct local mean estimation within each sub-cube (which additionally requires a strictly stronger membership querying oracle). Our results show that, in both passive and active settings, one can learn *globally* with a set of neural networks and achieve near minimax optimal label complexities.

3.3 Deep Active Learning with Abstention: Exponential Speedups

While the theoretical guarantees provided in Section 3.2 are near minimax optimal, the label complexity scales as $\text{poly}(\frac{1}{\varepsilon})$, which doesn't match the great empirical performance observed in deep active learning. In this section, we fill in this gap by leveraging the idea of abstention and provide a deep active learning algorithm that

achieves exponential label savings. We introduce the concepts of abstention and Chow's excess error in Section 3.3.1, and provide our label complexity guarantees in Section 3.3.2.

3.3.1 Active Learning without Low Noise Conditions

The previous section analyzes active learning under Tsybakov noise, which has been extensively studied in the literature since Castro and Nowak (2008). More recently, promising results are observed in active learning under Chow's excess error, but otherwise *without any low noise assumption* (Puchkin and Zhivotovskiy, 2021; Zhu and Nowak, 2022b). We introduce this setting in the following.

Abstention and Chow's error (Chow, 1970). We consider classifier of the form $\widehat{h} : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ where $\perp$ denotes the action of abstention. For any fixed $0 < \gamma < \frac{1}{2}$, the Chow's error is defined as

$$\mathrm{err}_\gamma(\widehat{h}) := \mathbb{P}_{(x,y)\sim\mathcal{D}_{xy}}(\widehat{h}(x) \neq y, \widehat{h}(x) \neq \perp) + (1/2 - \gamma) \cdot \mathbb{P}_{(x,y)\sim\mathcal{D}_{xy}}(\widehat{h}(x) = \perp).$$

The parameter γ can be chosen as a small constant, e.g., $\gamma = 0.01$, to avoid excessive abstention: The price of abstention is only marginally smaller than random guess (which incurs cost 0.5). The *Chow's excess error* is then defined as $\mathrm{excess}_\gamma(\widehat{h}) := \mathrm{err}_\gamma(\widehat{h}) - \mathrm{err}(h^\star)$ (Puchkin and Zhivotovskiy, 2021).

At a high level, analyzing with Chow's excess error allows slackness in predications of hard examples (e.g., data points whose $\eta(x)$ is close to $\frac{1}{2}$) by leveraging the power of abstention. Puchkin and Zhivotovskiy (2021); Zhu and Nowak (2022b) show that $\mathrm{polylog}(\frac{1}{\varepsilon})$ is always achievable in the *parametric* settings. We generalize their results to the *nonparametric* setting and analyze active learning with a set of neural networks.

3.3.2 Exponential Speedups with Abstention

In this section, we work with a set of neural network *regression functions* $\mathcal{F}_{\mathsf{dnn}} : \mathcal{X} \to [0, 1]$ (that approximates η) and then *construct* classifiers $h : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ *with an additional abstention action*. To work with a set of regression functions $\mathcal{F}_{\mathsf{dnn}}$, we analyze its "complexity" from the lenses of *pseudo dimension* $\mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}})$ (Pollard, 1984; Haussler, 1989, 1995) and *value function disagreement coefficient* $\theta^{\mathrm{val}}_{\mathcal{F}_{\mathsf{dnn}}}(\iota)$ (for some $\iota > 0$) (Foster et al., 2020c). We defer detailed definitions of these complexity measures to Section 3.7.1.

Algorithm 5 NeuralCAL++

Input: Accuracy level $\varepsilon \in (0,1)$, confidence level $\delta \in (0,1)$, abstention parameter $\gamma \in (0,1/2)$.

Let $\mathcal{F}_{\mathsf{dnn}}$ be a set of neural network regression functions obtained by (i) applying Theorem 3.4 with an appropriate approximation level κ (which satisfies $\frac{1}{\kappa} = \mathrm{poly}(\frac{1}{\gamma})\,\mathrm{polylog}(\frac{1}{\varepsilon\,\gamma})$), and (ii) applying a preprocessing step on the set of neural networks obtained from step (i). See Section 3.8.3 for details.

Define $T := \frac{\theta^{\mathrm{val}}_{\mathcal{F}_{\mathsf{dnn}}}(\gamma/4)\cdot \mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}})}{\varepsilon\,\gamma}$, $M := \lceil \log_2 T \rceil$, and $C_\delta := O(\mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}}) \cdot \log(T/\delta))$.

Define $\tau_m := 2^m$ for $m \geqslant 1$, $\tau_0 := 0$, and $\beta_m := 3(M-m+1)C_\delta$.

Define $\widehat{R}_m(f) := \sum_{t=1}^{\tau_{m-1}} Q_t(\widehat{f}(x_t) - y_t)^2$ with the convention that $\sum_{t=1}^{0} \ldots = 0$.

for epoch $m = 1, 2, \ldots, M$ **do**

  Get $\widehat{f}_m := \arg\min_{f \in \mathcal{F}_{\mathsf{dnn}}} \sum_{t=1}^{\tau_{m-1}} Q_t(f(x_t) - y_t)^2$.

  (Implicitely) Construct active set $\mathcal{F}_m := \left\{ f \in \mathcal{F}_{\mathsf{dnn}} : \widehat{R}_m(f) \leqslant \widehat{R}_m(\widehat{f}_m) + \beta_m \right\}$.

  Construct classifier $\widehat{h}_m : \mathcal{X} \to \{+1, -1, \perp\}$ as

$$\widehat{h}_m(x) := \begin{cases} \perp, & \text{if } [\mathsf{lcb}(x;\mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x;\mathcal{F}_m) + \frac{\gamma}{4}] \subseteq \left[\frac{1}{2} - \gamma, \frac{1}{2} + \gamma\right]; \\ \mathrm{sign}(2\widehat{f}_m(x) - 1), & \text{o.w.} \end{cases}$$

  and query function $g_m(x) := \mathbb{1}\left(\frac{1}{2} \in \left(\mathsf{lcb}(x;\mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x;\mathcal{F}_m) + \frac{\gamma}{4}\right)\right) \cdot \mathbb{1}(\widehat{h}_m(x) \neq \perp)$.

  if epoch $m = M$ **then**

    Return classifier $\widehat{h}_M$.

  for time $t = \tau_{m-1} + 1, \ldots, \tau_m$ **do**

    Observe $x_t \sim \mathcal{D}_X$. Set $Q_t := g_m(x_t)$.

    if $Q_t = 1$ **then**

      Query the label y_t of x_t.

We now present NeuralCAL++ (Algorithm 5), a deep active learning algorithm that leverages the power of abstention. NeuralCAL++ first initialize a set of set of neural network regression functions $\mathcal{F}_{\mathsf{dnn}}$ by applying a preprocessing step on top of the set of regression functions obtained from Theorem 3.4 with a carefully chosen

approximation level κ. The preprocessing step mainly contains two actions: (1) clipping $f_{\mathsf{dnn}} : \mathcal{X} \to \mathbb{R}$ into $\check{f}_{\mathsf{dnn}} : \mathcal{X} \to [0,1]$ (since we obviously have $\eta(x) \in [0,1]$); and (2) filtering out $f_{\mathsf{dnn}} \in \mathcal{F}_{\mathsf{dnn}}$ that are clearly not a good approximation of η. After initialization, NeuralCAL++ runs in epochs of geometrically increasing lengths. At the beginning of epoch $m \in [M]$, NeuralCAL++ (implicitly) constructs an active set of regression functions $\mathcal{F}_m$ that are "close" to the true conditional probability η. For any $x \sim \mathcal{D}_{\mathcal{X}}$, NeuralCAL++ constructs a lower bound $\mathsf{lcb}(x; \mathcal{F}_m) := \inf_{f \in \mathcal{F}_m} f(x)$ and an upper bound $\mathsf{ucb}(x; \mathcal{F}_m) := \sup_{f \in \mathcal{F}_m} f(x)$ as a confidence range of $\eta(x)$ (based on $\mathcal{F}_m$). An empirical classifier with an abstention option $\widehat{h}_m : \mathcal{X} \to \{+1, -1, \perp\}$ and a query function $g_m : \mathcal{X} \to \{0,1\}$ are then constructed based on the confidence range (and the abstention parameter γ). For any time step t within epoch m, NeuralCAL++ queries the label of the observed data point x_t if and only if $Q_t := g_m(x_t) = 1$. NeuralCAL++ returns $\widehat{h}_M$ as the learned classifier.

NeuralCAL++ is adapted from the algorithm developed in Zhu and Nowak (2022b), but with novel extensions. In particular, the algorithm presented in Zhu and Nowak (2022b) requires the existence of a $\bar{f} \in \mathcal{F}$ such that $\|\bar{f} - \eta\|_\infty \leqslant \varepsilon$ (to achieve ε Chow's excess error), Such an approximation requirement directly leads to $\mathrm{poly}(\frac{1}{\varepsilon})$ label complexity *in the nonparametric setting*, which is unacceptable. The initialization step of NeuralCAL++ (line 1) is carefully chosen to ensure that $\mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}}), \theta^{\mathrm{val}}_{\mathcal{F}_{\mathsf{dnn}}}(\frac{\gamma}{4}) = \mathrm{poly}(\frac{1}{\gamma}) \cdot \mathrm{polylog}(\frac{1}{\varepsilon})$; together with a sharper analysis of concentration results, these conditions help us derive the following deep active learning guarantees (also see Section 3.7 for a more general guarantee).

Theorem 3.10. *Fix any $\varepsilon, \delta, \gamma > 0$. With probability at least $1 - \delta$, Algorithm 5 (with an appropriate initialization at line 1) returns a classifier $\widehat{h}$ with Chow's excess error $\widetilde{O}(\varepsilon)$ after querying $\mathrm{poly}(\frac{1}{\gamma}) \cdot \mathrm{polylog}(\frac{1}{\varepsilon\delta})$ labels.*

We discuss two important aspects of Algorithm 5/Theorem 3.10 in the following, i.e., exponential savings and computational efficiency. We defer more detailed discussions to Section 3.8.4.1 and Section 3.8.4.2.

- **Exponential speedups.** Theorem 3.10 shows that, equipped with an abstention option, deep active learning enjoys $\mathrm{polylog}(\frac{1}{\varepsilon})$ label complexity. This

provides theoretical justifications for great empirical results of deep active learning observed in practice. Moreover, Algorithm 5 outputs a classifier that abstains *properly*, i.e., it abstains only if abstention is the optimal choice; such a property further implies $\mathrm{polylog}(\frac{1}{\varepsilon})$ label complexity under *standard* excess error and Massart noise (Massart and Nédélec, 2006).

- **Computational efficiency.** Suppose one can efficiently implement a (weighted) square loss regression oracle over the *initialized* set of neural networks $\mathcal{F}_{\mathsf{dnn}}$: Given any set $\mathcal{S}$ of weighted examples $(w, x, y) \in \mathbb{R}_+ \times \mathcal{X} \times \mathcal{Y}$ as input, the regression oracle outputs $\widehat{f}_{\mathsf{dnn}} := \arg\min_{f \in \mathcal{F}_{\mathsf{dnn}}} \sum_{(w,x,y)\in\mathcal{S}} w(f(x) - y)^2$.[4] Algorithm 5 can then be *efficiently* implemented with $\mathrm{poly}(\frac{1}{\gamma}) \cdot \frac{1}{\varepsilon}$ oracle calls.

While the label complexity obtained in Theorem 3.10 has desired dependence on $\mathrm{polylog}(\frac{1}{\varepsilon})$, its dependence on γ can be of order $\gamma^{-\mathrm{poly}(d)}$. Our next result shows that, however, such dependence is unavoidable even in the case of learning a single ReLU function.

Theorem 3.11. *Fix any $\gamma \in (0, 1/8)$. For any accuracy level ε sufficiently small, there exists a problem instance such that (1) $\eta \in \mathcal{W}_1^{1,\infty}(\mathcal{X})$ and is of the form $\eta(x) := \mathsf{ReLU}(\langle w, x\rangle + a) + b$; and (2) for any active learning algorithm, it takes at least $\gamma^{-\Omega(d)}$ labels to identify an ε-optimal classifier, for either standard excess error or Chow's excess error (with parameter γ).*

3.4 Extensions

Previous results are developed in the commonly studied Sobolev/Hölder spaces. Our techniques, however, are generic and can be adapted to other function spaces, given neural network approximation results. In this section, we provide extensions of our results to the Radon BV^2 space, which was recently proposed as the natural

[4] In practice, one can approximate this oracle by running stochastic gradient descent.

function space associated with ReLU neural networks (Ongie et al., 2020; Parhi and Nowak, 2021, 2022b,a; Unser, 2022).[5]

The Radon BV^2 space. The Radon BV^2 unit ball over domain $\mathcal{X}$ is defined as $\mathscr{R}\,\mathsf{BV}_1^2(\mathcal{X}) := \{f : \|f\|_{\mathscr{R}\mathsf{BV}^2(\mathcal{X})} \leqslant 1\}$, where $\|f\|_{\mathscr{R}\mathsf{BV}^2(\mathcal{X})}$ denotes the Radon BV^2 norm of f over domain $\mathcal{X}$.[6] Following Parhi and Nowak (2022a), we assume $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \leqslant 1\}$ and $\eta \in \mathscr{R}\,\mathsf{BV}_1^2(\mathcal{X})$.

The Radon BV^2 space naturally contains neural networks of the form $f_{\mathsf{dnn}}(x) = \sum_{k=1}^{K} v_i \cdot \mathsf{ReLU}(w_i^\top x + b_i)$. On the contrary, such f_{dnn} doesn't lie in any Sobolev space of order $\alpha \geqslant 2$ (since f_{dnn} doesn't have second order *weak* derivative). Thus, if η takes the form of the aforementioned neural network (e.g., $\eta = f_{\mathsf{dnn}}$), approximating η up to κ from a Sobolev perspective requires $\widetilde{O}(\kappa^{-d})$ total parameters, which suffers from the curse of dimensionality. On the other side, however, such bad dependence on dimensionality goes away when approximating from a Radon BV^2 perspective, as shown in the following theorem.

Theorem 3.12 (Parhi and Nowak (2022a))**.** *Fix any $\kappa > 0$. For any $f^\star \in \mathscr{R}\,\mathsf{BV}_1^2(\mathcal{X})$, there exists a one-hidden layer neural network f_{dnn} of width $K = O(\kappa^{-\frac{2d}{d+3}})$ such that $\|f^\star - f_{\mathsf{dnn}}\|_\infty \leqslant \kappa$.*

Equipped with this approximation result, we provide the active learning guarantees for learning a smooth function within the Radon BV^2 unit ball as follows.

Theorem 3.13. *Suppose $\eta \in \mathscr{R}\,\mathsf{BV}_1^2(\mathcal{X})$ and the Tsybakov noise condition is satisfied with parameter $\beta \geqslant 0$. Fix any $\varepsilon, \delta > 0$. There exists an algorithm such that, with probability at least $1 - \delta$, it learns a classifier $\widehat{h} \in \mathcal{H}_{\mathsf{dnn}}$ with excess error $\widetilde{O}(\varepsilon)$ after querying $\widetilde{O}(\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) \cdot \varepsilon^{-\frac{4d+6}{(1+\beta)(d+3)}})$ labels.*

Compared to the label complexity obtained in Theorem 3.9, the label complexity obtained in the above theorem doesn't suffer from the curse of dimensionality: For

[5] Other extensions are also possible given neural network approximation results, e.g., recent results established in Lu et al. (2021).

[6] We provide more mathematical backgrounds and associated definitions in Section 3.8.5.

d large enough, the above label complexity scales as $\varepsilon^{-O(1)}$ yet label complexity in Theorem 3.9 scales as $\varepsilon^{-O(d)}$. Active learning guarantees under Chow's excess error in the Radon BV^2 space are similar to results presented in Theorem 3.10, and are thus deferred to Section 3.8.5.

3.5 Discussion

We provide the first near-optimal deep active learning guarantees, under both standard excess error and Chow's excess error. Our results are powered by generic algorithms and analyses developed for active learning that bridge approximation guarantees into label complexity guarantees. We outline some natural directions for future research below.

- **Disagreement coefficients for neural networks.** While we have provided some results regarding the disagreement coefficients for neural networks, we believe a comprehensive investigation on this topic is needed. For instance, can we discover more general settings where the classifier-based disagreement coefficient can be upper bounded by $O(1)$? It is also interesting to explore sharper analyses on the value function disagreement coefficient.
- **Adaptivity in deep active learning.** Our current results are established with the knowledge of some problem-dependent parameters, e.g., the smoothness parameters regarding the function spaces and the noise levels. It will be interesting to see if one can develop algorithms that can automatically adapt to unknown parameters, e.g., by leveraging techniques developed in Locatelli et al. (2017, 2018).

3.6 Generic Version of Algorithm 4 and Its Guarantees

We present Algorithm 6 below, a generic version of Algorithm 4 that doesn't require the approximating classifiers to be neural networks. The guarantees of Algorithm 6 are provided in Theorem 3.14, which is proved in Section 3.6.2 based on supporting lemmas provided in Section 3.6.1.

Algorithm 6 RobustCAL with Approximation

Input: Accuracy level $\varepsilon \in (0,1)$, confidence level $\delta \in (0,1)$.

Let $\mathcal{H}$ be a set of approximating classifiers such that $\inf_{h\in\mathcal{H}} \mathrm{err}(h) - \mathrm{err}(h^\star) = O(\varepsilon)$.

Define $T := \varepsilon^{-\frac{2+\beta}{1+\beta}} \cdot \mathrm{VCdim}(\mathcal{H})$, $M := \lceil \log_2 T \rceil$, $\tau_m := 2^m$ for $m \geqslant 1$ and $\tau_0 := 0$.

Define $\rho_m := O\left(\left(\frac{\mathrm{VCdim}(\mathcal{H})\cdot\log(\tau_{m-1})\cdot\log(M/\delta)}{\tau_{m-1}}\right)^{\frac{1+\beta}{2+\beta}}\right)$ for $m \geqslant 2$ and $\rho_1 := 1$.

Define $\widehat{R}_m(h) := \sum_{t=1}^{\tau_{m-1}} Q_t \mathbb{1}(h(x_t) \neq y_t)$ with the convention that $\sum_{t=1}^{0} \ldots = 0$.

Initialize $\mathcal{H}_0 := \mathcal{H}$.

for epoch $m = 1, 2, \ldots, M$ **do**

  Update active set $\mathcal{H}_m := \left\{h \in \mathcal{H}_{m-1} : \widehat{R}_m(h) \leqslant \inf_{h\in\mathcal{H}_{m-1}} \widehat{R}_m(h) + \tau_{m-1}\cdot\rho_m\right\}$

  if epoch $m = M$ **then**

    Return any classifier $\widehat{h} \in \mathcal{H}_M$.

  for time $t = \tau_{m-1}+1, \ldots, \tau_m$ **do**

    Observe $x_t \sim \mathcal{D}_X$. Set $Q_t := \mathbb{1}(x_t \in \mathsf{DIS}(\mathcal{H}_m))$.

    if $Q_t = 1$ **then**

      Query the label y_t of x_t.

We provide guarantees for Algorithm 6, and then specialize them to the settings with neural network approximation, i.e., in Theorem 3.9 and Theorem 3.13. As discussed before, our analysis is based on the analysis RobustCAL, but with novel extensions in removing the requirements that the Bayes classifier is in the class (or a Bernstein condition for every $h \in \mathcal{H}$).

Theorem 3.14. *Fix $\varepsilon, \delta > 0$. With probability at least $1-\delta$, Algorithm 6 returns a classifier $\widehat{h} \in \mathcal{H}$ with excess error $\widetilde{O}(\varepsilon)$ after querying*

$$\widetilde{O}\left(\theta_{\mathcal{H}}(\varepsilon^{\frac{\beta}{1+\beta}}) \cdot \varepsilon^{-\frac{2}{1+\beta}} \cdot \mathrm{VCdim}(\mathcal{H})\right)$$

labels.

3.6.1 Supporting Lemmas

We first recall that Tsybakov noise condition leads to the so-called Bernstein condition (with respect to Bayes classifier $h^\star$).

Lemma 3.15 (Tsybakov (2004))**.** *Suppose $\mathcal{D}_{XY}$ satisfies the Tsybakov noise condition with parameter $\beta \geqslant 0$, then there exists an universal constant $c' > 0$ such that we have*

$$\mathbb{P}_{x\sim\mathcal{D}_X}(h(x) \neq h^\star(x)) \leqslant c'(\mathrm{err}(h) - \mathrm{err}(h^\star))^{\frac{\beta}{1+\beta}}$$

for any $h : \mathcal{X} \to \mathcal{Y}$.

We next present a lemma in the passive learning setting, which will later be incorporated into the active learning setting. We first define some notations. Suppose $D_n = \{(x_i, y_i)\}_{i=1}^n$ are n i.i.d. data points drawn from $\mathcal{D}_{XY}$. For any $h : \mathcal{X} \to \mathcal{Y}$, we denote $\bar{R}_n(h) := \sum_{i=1}^n \mathbb{1}(h(x_i) \neq y_i)$ as the empirical error of h over dataset D_n. We clearly have $\mathbb{E}[\bar{R}_n(h)] = n \cdot \mathrm{err}(h)$ by i.i.d. assumption.

Lemma 3.16. *Fix $\varepsilon, \bar{\delta} > 0$. Suppose $\mathcal{D}_{XY}$ satisfies Tsybakov noise condition with parameter $\beta \geqslant 0$ and $\mathrm{err}(\check{h}) - \mathrm{err}(h^\star) = O(\varepsilon)$, where $\check{h} = \arg\max_{h\in\mathcal{H}} \mathrm{err}(h)$ and $h^\star$ is the Bayes classifier. Let $D_n = \{(x_i, y_i)\}_{i=1}^n$ be a set of n i.i.d. data points drawn from $\mathcal{D}_{XY}$. If $\beta > 0$, suppose n satisfies*

$$n \leqslant \varepsilon^{-\frac{2+\beta}{1+\beta}} \cdot \mathrm{VCdim}(\mathcal{H})^{\frac{2+2\beta}{\beta}} \cdot \log(\bar{\delta}^{-1}) \cdot (\log n)^{\frac{2+2\beta}{\beta}}.$$

With probability at least $1-\bar{\delta}$*, we have the following inequalities hold:*

$$n\cdot(\mathrm{err}(h)-\mathrm{err}(h^\star))\leqslant 2\cdot(\bar{R}_n(h)-\bar{R}_n(\check{h}))+n\cdot\rho(n,\bar{\delta}),\quad \forall h\in\mathcal{H}, \tag{3.1}$$

$$\bar{R}_n(\check{h})-\min_{h\in\mathcal{H}}\bar{R}_n(h)\leqslant n\cdot\rho(n,\bar{\delta}), \tag{3.2}$$

where $\rho(n,\bar{\delta})=C\cdot\left(\left(\frac{\mathrm{VCdim}(\mathcal{H})\cdot\log n\cdot\log\bar{\delta}^{-1}}{n}\right)^{\frac{1+\beta}{2+\beta}}+\varepsilon\right)$ *with a universal constant* $C>0$.[7]

Proof. Denote $\overline{\mathcal{H}}:=\mathcal{H}\cup\{h^\star\}$. We know that $\mathrm{VCdim}(\overline{\mathcal{H}})\leqslant\mathrm{VCdim}(\mathcal{H})+1=O(\mathrm{VCdim}(\mathcal{H}))$. From Lemma 3.15, we know Bernstein condition is satisfied with respect to $\overline{\mathcal{H}}$ and $h^\star\in\overline{\mathcal{H}}$. Invoking Lemma 3.1 in Hanneke (2014), with probability at least $1-\frac{\bar{\delta}}{2}$, $\forall h\in\overline{\mathcal{H}}$, we have

$$n\cdot(\mathrm{err}(h)-\mathrm{err}(h^\star))\leqslant\max\{2\cdot(\bar{R}_n(h)-\bar{R}_n(h^\star)),n\cdot\bar{\rho}(n,\bar{\delta})\}, \tag{3.3}$$

$$\bar{R}_n(h)-\min_{h\in\overline{\mathcal{H}}}\bar{R}_n(h)\leqslant\max\{2n\cdot(\mathrm{err}(h)-\mathrm{err}(h^\star)),n\cdot\bar{\rho}(n,\bar{\delta})\}, \tag{3.4}$$

where $\bar{\rho}(n,\bar{\delta})=O\left(\left(\frac{\mathrm{VCdim}(\overline{\mathcal{H}})\cdot\log n+\log\bar{\delta}^{-1}}{n}\right)^{\frac{1+\beta}{2+\beta}}\right)=O\left(\left(\frac{\mathrm{VCdim}(\mathcal{H})\cdot\log n\cdot\log\bar{\delta}^{-1}}{n}\right)^{\frac{1+\beta}{2+\beta}}\right)$.

Eq. (3.2) follows by taking $h=\check{h}$ in Eq. (3.4) and noticing that

$$\begin{aligned}\bar{R}_h(\check{h})-\min_{h\in\mathcal{H}}\bar{R}_n(h)&\leqslant\bar{R}_n(\check{h})-\min_{h\in\overline{\mathcal{H}}}\bar{R}_n(h)\\&\leqslant\max\{2n\cdot O(\varepsilon),n\cdot\bar{\rho}(n,\bar{\delta})\},\end{aligned}$$

where we use the assumption that $\mathrm{err}(\check{h})-\mathrm{err}(h^\star)=O(\varepsilon)$.

To derive Eq. (3.1), we first notice that applying Eq. (3.3) for any $h\in\mathcal{H}$, we have

$$n\cdot(\mathrm{err}(h)-\mathrm{err}(h^\star))\leqslant 2\cdot(\bar{R}_n(h)-\bar{R}_n(\check{h})+\bar{R}_n(\check{h})-\bar{R}_n(h^\star))+n\cdot\bar{\rho}(n,\bar{\delta}).$$

[7] The logarithmic factors in this bound might be further optimized. We don't focus on optimizing logarithmic factors.

We next only need to upper bound $\bar{R}_n(\check{h}) - \bar{R}_n(h^\star)$, and show that it is order-wise smaller than $n \cdot \rho(n, \bar{\delta})$. We consider random variable $g_i := \mathbb{1}(\check{h}(x_i) \neq y_i) - \mathbb{1}(h^\star(x_i) \neq y_i)$. We have

$$\begin{aligned}\mathbb{V}(g_i) &\leqslant \mathbb{E}[g_i^2] \\ &= \mathbb{E}[\mathbb{1}(\check{h}(x_i) \neq h^\star(x_i)] \\ &= O\left(\varepsilon^{\frac{\beta}{1+\beta}}\right),\end{aligned}$$

where the last line follows from Lemma 3.15 and the assumption that $\mathrm{err}(\check{h}) - \mathrm{err}(h^\star) = O(\varepsilon)$. Denote $g = \frac{1}{n}\sum_{i=1}^n g_i = \frac{1}{n}(\bar{R}_n(\check{h}) - \bar{R}_n(h^\star))$, and notice that $\mathbb{E}[g] = \mathrm{err}(\check{h}) - \mathrm{err}(h^\star)$. Applying Bernstein inequality on $-g$, with probability at least $1 - \frac{\bar{\delta}}{2}$, we have

$$g - \mathbb{E}[g] \leqslant O\left(\left(\frac{\varepsilon^{\frac{\beta}{1+\beta}} \log \bar{\delta}^{-1}}{n}\right)^{\frac{1}{2}} + \frac{\log \bar{\delta}^{-1}}{n}\right),$$

which further leads to

$$\bar{R}_n(\check{h}) - \bar{R}_n(h^\star) \leqslant n \cdot O\left(\varepsilon + \left(\frac{\varepsilon^{\frac{\beta}{1+\beta}} \log \bar{\delta}^{-1}}{n}\right)^{\frac{1}{2}} + \frac{\log \bar{\delta}^{-1}}{n}\right).$$

The RHS is order-wise smaller than ρ_n when $\beta = 0$. We consider the case when $\beta > 0$ next. Since $\log(\bar{\delta}^{-1})/n$ is clearly a lower-order term compared to ρ_n, we only need to show that $\left(\frac{\varepsilon^{\frac{\beta}{1+\beta}} \log \bar{\delta}^{-1}}{n}\right)^{\frac{1}{2}}$ is order-wise smaller than ρ_n. We can easily check that

$$\left(\frac{\varepsilon^{\frac{\beta}{1+\beta}} \log \bar{\delta}^{-1}}{n}\right)^{\frac{1}{2}} \leqslant \left(\frac{\mathrm{VCdim}(\mathcal{H}) \cdot \log n \cdot \log \bar{\delta}^{-1}}{n}\right)^{\frac{1+\beta}{2+\beta}}$$

whenever n satisfies the following condition

$$n \leqslant \varepsilon^{-\frac{2+\beta}{1+\beta}} \cdot \mathrm{VCdim}(\mathcal{H})^{\frac{2+2\beta}{\beta}} \cdot \log(\bar{\delta}^{-1}) \cdot (\log n)^{\frac{2+2\beta}{\beta}}.$$

□

We denote $\check{h} = \arg\min_{h \in \mathcal{H}} \mathrm{err}(h)$. By assumption of Theorem 3.14, we have $\mathrm{err}(\check{h}) - \mathrm{err}(h^\star) = O(\varepsilon)$. For any $h \in \mathcal{H}$, we also use the shorthand $\bar{R}_m(h) = \bar{R}_{\tau_{m-1}}(h) := \sum_{t=1}^{\tau_{m-1}} \mathbb{1}(h(x_t) \neq y_t)$. Note that $\bar{R}_m$ is only used in analysis since some y_t are not observable.

Lemma 3.17. *With probability at least $1 - \frac{\delta}{2}$, the following holds true for all epochs $m \in [M]$:*

1. $\check{h} \in \mathcal{H}_m$.
2. $\mathrm{err}(h) - \mathrm{err}(h^\star) \leqslant 3\rho_m, \forall h \in \mathcal{H}_m$.

Proof. For each $m = 2, 3, \ldots, M$, we invoke Lemma 3.16 with $n = \tau_{m-1}$ and $\bar{\delta} = \delta/2M$, which guarantees that

$$\tau_{m-1} \cdot (\mathrm{err}(h) - \mathrm{err}(h^\star)) \leqslant 2 \cdot (\bar{R}_m(h) - \bar{R}_m(\check{h})) + \tau_{m-1} \cdot \rho_m, \quad \forall h \in \mathcal{H}, \tag{3.5}$$

$$\bar{R}_m(\check{h}) - \min_{h \in \mathcal{H}} \bar{R}_m(h) \leqslant \tau_{m-1} \cdot \rho_m. \tag{3.6}$$

Note that the choice T chosen in Algorithm 6 clearly satisfies the requirement needed (for $n = \tau_{m-1}$) in Lemma 3.16 when $\beta > 0$; and ensures that the second term in $\rho(\tau_{m-1}, \delta/2M)$ (i.e., ε, see Lemma 3.16 for definition of $\rho(\tau_{m-1}, \delta/2M)$) is a lower-order term compared to the first term.

We use $\mathcal{E}$ to denote the good event where Eq. (3.5) and Eq. (3.6) hold true across $m = 2, 3, \ldots, M$. This good event happens with probability at least $1 - \frac{\delta}{2}$. We analyze under $\mathcal{E}$ in the following.

We prove Lemma 3.17 through induction. The statements clearly hold true for $m = 1$. Suppose the statements hold true up to epoch m, we next prove the correctness for epoch $m + 1$.

We know that $\check{h} \in \mathcal{H}_m$ by assumption. Based on the querying criteria of Algorithm 6, we know that

$$\widehat{R}_{m+1}(\check{h}) - \widehat{R}_{m+1}(h) = \overline{R}_{m+1}(\check{h}) - \overline{R}_{m+1}(h), \quad \forall h \in \mathcal{H}_m \tag{3.7}$$

From Eq. (3.6), we also have

$$\begin{aligned}\overline{R}_{m+1}(\check{h}) - \min_{h \in \mathcal{H}_m} R_{m+1}(h) &\leqslant \overline{R}_{m+1}(\check{h}) - \min_{h \in \mathcal{H}} R_{m+1}(h) \\ &\leqslant \tau_m \cdot \rho_{m+1}.\end{aligned}$$

Combining the above two inequalities shows that

$$\widehat{R}_{m+1}(\check{h}) - \widehat{R}_{m+1}(h) \leqslant \tau_m \cdot \rho_{m+1},$$

implying that $\check{h} \in \mathcal{H}_{m+1}$ (due to the construction of $\mathcal{H}_{m+1}$ in Algorithm 6).

Based on Eq. (3.7), the construction $\mathcal{H}_{m+1}$ and the fact that $\check{h} \in \mathcal{H}_m$, we know that, for any $h \in \mathcal{H}_{m+1} \subseteq \mathcal{H}_m$,

$$\begin{aligned}\overline{R}_{m+1}(h) - \overline{R}_{m+1}(\check{h}) &= \widehat{R}_{m+1}(h) - \widehat{R}_{m+1}(\check{h}) \\ &\leqslant \widehat{R}_{m+1}(h) - \min_{h \in \mathcal{H}_m} \widehat{R}_{m+1}(h) \\ &\leqslant \tau_m \cdot \rho_{m+1}.\end{aligned}$$

Plugging the above inequality into Eq. (3.5) (at epoch $m+1$) leads to $\mathrm{err}(h) - \mathrm{err}(h^\star) \leqslant 3\rho_{m+1}$ for any $h \in \mathcal{H}_{m+1}$. We thus prove the desired statements at epoch $m+1$. □

3.6.2 Proof of Theorem 3.14

Theorem 3.14. *Fix $\varepsilon, \delta > 0$. With probability at least $1-\delta$, Algorithm 6 returns a classifier $\widehat{h} \in \mathcal{H}$ with excess error $\widetilde{O}(\varepsilon)$ after querying*

$$\widetilde{O}\left(\theta_{\mathcal{H}}(\varepsilon^{\frac{\beta}{1+\beta}}) \cdot \varepsilon^{-\frac{2}{1+\beta}} \cdot \mathrm{VCdim}(\mathcal{H})\right)$$

labels.

Proof. Based on Lemma 3.17, we know that, with probability at least $1 - \frac{\delta}{2}$, we have

$$\begin{aligned}\operatorname{err}(\widehat{h}) - \operatorname{err}(h^\star) &\leqslant 3\rho_M \\ &= O\left(\left(\frac{\operatorname{VCdim}(\mathcal{H}) \cdot \log(\tau_{M-1}) \cdot \log(M/\delta)}{\tau_{M-1}}\right)^{\frac{1+\beta}{2+\beta}}\right) \\ &= \widetilde{O}(\varepsilon),\end{aligned}$$

where we use the definition of T and τ_M.

We next analyze the label complexity of Algorithm 6. Since Algorithm 6 stops and the beginning at epoch M, we only need to calculated the label complexity in the first $M-1$ epochs. We have

$$\begin{aligned}\sum_{t=1}^{\tau_{M-1}} Q_t &= \sum_{m=1}^{M-1} (\tau_m - \tau_{m-1}) \cdot \mathbb{1}(x_t \in \mathsf{DIS}(\mathcal{H}_m)) \\ &\leqslant \sum_{m=1}^{M-1} (\tau_m - \tau_{m-1}) \cdot \mathbb{1}\left(x_t \in \mathsf{DIS}(\mathcal{B}_{\mathcal{H}}(h^\star, c'(3\rho_m)^{\frac{\beta}{1+\beta}}))\right),\end{aligned}$$

where on the last line we use the facts (1) $\operatorname{err}(h) - \operatorname{err}(h^\star) \leqslant 3\rho_m, \forall h \in \mathcal{H}_m$ from Lemma 3.17; and (2) $\mathbb{P}(x : h(x) \neq h^\star(x)) \leqslant c'(\operatorname{err}(h) - \operatorname{err}(h^\star))^{\frac{\beta}{1+\beta}}$ from Lemma 3.16 (with the same constant c'). Suppose $\operatorname{err}(\check{h}) - \operatorname{err}(h^\star) = c''\varepsilon$ (with another universal constant c'' by assumption). Applying Lemma 3.16 on $\check{h}$ leads to the fact that $h^\star \in \mathcal{B}_{\mathcal{H}}(\check{h}, c'(c''\varepsilon)^{\frac{\beta}{1+\beta}})$. Since $\mathbb{P}(x : h(x) \neq \check{h}(x)) \leqslant \mathbb{P}(x : h(x) \neq h^\star(x)) + \mathbb{P}(x : h^\star(x) \neq \check{h}(x))$, we further have

$$\sum_{t=1}^{\tau_{M-1}} Q_t \leqslant \sum_{m=1}^{M-1} (\tau_m - \tau_{m-1}) \cdot \mathbb{1}\left(x_t \in \mathsf{DIS}(\mathcal{B}_{\mathcal{H}}(\check{h}, \bar{c} \cdot \rho_m{}^{\frac{\beta}{1+\beta}}))\right),$$

with a universal constant $\bar{c} > 0$. Noticing that the RHS is a sum of independent Bernoulli random variables, applying a Bernstein-type bound (e.g., Lemma 3.23),

on a good event $\mathcal{E}'$ that happens with probability at least $1 - \frac{\delta}{2}$, we have

$$\begin{aligned}\sum_{t=1}^{\tau_{M-1}} Q_t &\leqslant 2\sum_{m=1}^{M-1}(\tau_m - \tau_{m-1}) \cdot \mathbb{P}\Big(x \in \mathsf{DIS}(\mathcal{B}_{\mathcal{H}}(\check{h}, \bar{c} \cdot \rho_m{}^{\frac{\beta}{1+\beta}}))\Big) + 4\log(4/\delta) \\ &\leqslant 2\sum_{m=2}^{M-1} \tau_{m-1} \cdot \theta_{\mathcal{H},\check{h}}\left(\bar{c} \cdot \rho_m^{\frac{\beta}{1+\beta}}\right) \cdot \bar{c} \cdot \rho_m^{\frac{\beta}{1+\beta}} + 4\log(4/\delta) + 4 \\ &\leqslant 2M \cdot \theta_{\mathcal{H},\check{h}}\left(\bar{c} \cdot \rho_M^{\frac{\beta}{1+\beta}}\right) \cdot \left(\bar{c} \cdot \tau_{M-1} \cdot \rho_M^{\frac{\beta}{1+\beta}}\right) + 4\log(4/\delta) + 4,\end{aligned}$$

where the second using the definition of disagreement coefficient; and the last line follows from the fact that ρ_m is non-increasing and $\tau_{m-1} \cdot \rho_m$ is increasing. Basic algebra and basic properties of the disagreement coefficient (i.e., Theorem 7.1 and Corollary 7.2 in Hanneke (2014)) shows that

$$\sum_{t=1}^{\tau_{M-1}} Q_t \leqslant \widetilde{O}\Big(\theta_{\mathcal{H}}(\varepsilon^{\frac{\beta}{1+\beta}}) \cdot \varepsilon^{-\frac{2}{1+\beta}} \cdot \mathrm{VCdim}(\mathcal{H})\Big),$$

under event $\mathcal{E} \cap \mathcal{E}'$, which happen with probability at least $1 - \delta$. □

3.7 Generic Version of Algorithm 5 and Its Guarantees

This section is organized as follows. We first introduce some complexity measures in Section 3.7.1. We then provide the generic algorithm (Algorithm 7) and state its theoretical guarantees (Theorem 3.21) in Section 3.7.2. Some of the results have appeared in Chapter 2, and we provide them here for completeness.

3.7.1 Complexity Measures

We first introduce *pseudo dimension* (Pollard, 1984; Haussler, 1989, 1995), a complexity measure used to analyze real-valued functions.

Definition 3.18 (Pseudo dimension)**.** *Consider a set of real-valued function $\mathcal{F} : \mathcal{X} \to \mathbb{R}$. The pseudo dimension* $\mathrm{Pdim}(\mathcal{F})$ *of $\mathcal{F}$ is defined as the VC dimension of the set of threshold functions $\{(x, \zeta) \mapsto \mathbb{1}(f(x) > \zeta) : f \in \mathcal{F}\}$.*

As discussed in Bartlett et al. (2019), similar results as in Theorem 3.5 holds true for $\mathrm{Pdim}(\mathcal{F})$ as well.

Theorem 3.19 (Bartlett et al. (2019))**.** *Let $\mathcal{F}_{\mathsf{dnn}}$ be a set of neural network regression functions of the same architecture and with W parameters arranged in L layers. We then have*

$$\Omega(WL\log(W/L)) \leqslant \mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}}) \leqslant O(WL\log(W)).$$

We now introduce *value function disagreement coefficient*, which is proposed by Foster et al. (2020c) in contextual bandits and then adapted to active learning by Zhu and Nowak (2022b) with additional supreme over the marginal distribution $\mathcal{D}_X$ to deal with distributional shifts caused by selective sampling.

Definition 3.20 (Value function disagreement coefficient)**.** *For any $f^\star \in \mathcal{F}$ and $\gamma_0, \varepsilon_0 > 0$, the value function disagreement coefficient $\theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0)$ is defined as*

$$\sup_{\mathcal{D}_X} \sup_{\gamma > \gamma_0, \varepsilon > \varepsilon_0} \left\{ \frac{\gamma^2}{\varepsilon^2} \cdot \mathbb{P}_{\mathcal{D}_X}\big(\exists f \in \mathcal{F} : |f(x) - f^\star(x)| > \gamma, \|f - f^\star\|_{\mathcal{D}_X} \leqslant \varepsilon\big) \right\} \vee 1,$$

where $\|f\|^2_{\mathcal{D}_X} := \mathbb{E}_{x\sim\mathcal{D}_X}[f^2(x)]$. We also define $\theta^{\mathrm{val}}_{\mathcal{F}}(\gamma_0) := \sup_{f^\star\in\mathcal{F},\varepsilon_0>0} \theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma_0, \varepsilon_0)$.

3.7.2 The Generic Algorithm and Its Guarantees

We present Algorithm 7, a generic version of Algorithm 5 that doesn't require the approximating classifiers to be neural networks.

Algorithm 7 NeuralCAL++ (Generic Version)

Input: Accuracy level $\varepsilon \in (0,1)$, confidence level $\delta \in (0,1)$, abstention parameter $\gamma \in (0,1/2)$.

Let $\mathcal{F} : \mathcal{X} \to [0,1]$ be a set of regression functions such that there exists a regression function $\bar{f} \in \mathcal{F}$ with $\|\bar{f} - \eta\|_\infty \leqslant \kappa \leqslant \gamma/4$.

Define $T := \frac{\theta_{\mathcal{F}}^{\mathrm{val}}(\gamma/4) \cdot \mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}$, $M := \lceil \log_2 T \rceil$, and $C_\delta := O(\mathrm{Pdim}(\mathcal{F}) \cdot \log(T/\delta))$.

Define $\tau_m := 2^m$ for $m \geqslant 1$, $\tau_0 := 0$, and $\beta_m := 3(M - m + 1)C_\delta$.

Define $\widehat{R}_m(f) := \sum_{t=1}^{\tau_{m-1}} Q_t(\widehat{f}(x_t) - y_t)^2$ with the convention that $\sum_{t=1}^{0} \ldots = 0$.

for epoch $m = 1, 2, \ldots, M$ **do**

  Get $\widehat{f}_m := \arg\min_{f \in \mathcal{F}} \sum_{t=1}^{\tau_{m-1}} Q_t(f(x_t) - y_t)^2$.

  (Implicitely) Construct active set $\mathcal{F}_m := \left\{ f \in \mathcal{F} : \widehat{R}_m(f) \leqslant \widehat{R}_m(\widehat{f}_m) + \beta_m \right\}$.

  Construct classifier $\widehat{h}_m : \mathcal{X} \to \{+1, -1, \perp\}$ as

$$\widehat{h}_m(x) := \begin{cases} \perp, & \text{if } [\mathsf{lcb}(x; \mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x; \mathcal{F}_m) + \frac{\gamma}{4}] \subseteq \left[\frac{1}{2} - \gamma, \frac{1}{2} + \gamma\right]; \\ \mathrm{sign}(2\widehat{f}_m(x) - 1), & \text{o.w.} \end{cases}$$

  and query function $g_m(x) := \mathbb{1}\left(\frac{1}{2} \in \left(\mathsf{lcb}(x; \mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x; \mathcal{F}_m) + \frac{\gamma}{4}\right)\right) \cdot \mathbb{1}(\widehat{h}_m(x) \neq \perp)$.

  if epoch $m = M$ **then**

    Return classifier $\widehat{h}_M$.

  for time $t = \tau_{m-1} + 1, \ldots, \tau_m$ **do**

    Observe $x_t \sim \mathcal{D}_X$. Set $Q_t := g_m(x_t)$.

    if $Q_t = 1$ **then**

      Query the label y_t of x_t.

We next state the theoretical guarantees for Algorithm 7.

Theorem 3.21. *Suppose* $\theta_{\mathcal{F}}^{\mathrm{val}}(\gamma/4) \leqslant \bar{\theta}$ *and the approximation level* $\kappa \in (0, \gamma/4]$ *satisfies*

$$\left(\frac{432\bar{\theta} \cdot M^2}{\gamma^2}\right) \cdot \kappa^2 \leqslant \frac{1}{10}. \tag{3.8}$$

With probability at least $1 - \delta$, *Algorithm 7 returns a classifier* $\widehat{h} : \mathcal{X} \to \{+1, -1, \perp\}$ *with*

Chow's excess error

$$\mathsf{excess}_\gamma(\widehat{h}) = O\left(\varepsilon \cdot \log\left(\frac{\bar{\theta} \cdot \mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}\right)\right),$$

after querying at most

$$O\left(\frac{M^2 \cdot \mathrm{Pdim}(\mathcal{F}) \cdot \log(T/\delta) \cdot \bar{\theta}}{\gamma^2}\right)$$

labels.

Theorem 3.21 is proved in Section 3.7.3, based on supporting lemmas and theorems established in Section 3.7.2.1 and Section 3.7.2.2. The general result (Theorem 3.21) will be used to prove results in specific settings (e.g., Theorem 3.10 and Theorem 3.59).

3.7.2.1 Concentration Results

The Freedman's inequality is commonly used in the field of active learning and contextual bandits, e.g., (Freedman, 1975; Agarwal et al., 2014; Krishnamurthy et al., 2019; Foster et al., 2020c). We thus state the result without proof.

Lemma 3.22 (Freedman's inequality)**.** *Let $(X_t)_{t\leqslant T}$ be a real-valued martingale difference sequence adapted to a filtration $\mathfrak{F}_t$, and let $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot \mid \mathfrak{F}_{t-1}]$. If $|X_t| \leqslant B$ almost surely, then for any $\eta \in (0, 1/B)$ it holds with probability at least $1-\delta$,*

$$\sum_{t=1}^{T} X_t \leqslant \eta \sum_{t=1}^{T} \mathbb{E}_t[X_t^2] + \frac{\log \delta^{-1}}{\eta}.$$

Lemma 3.23. *Let $(X_t)_{t\leqslant T}$ to be real-valued sequence of random variables adapted to a filtration $\mathfrak{F}_t$. If $|X_t| \leqslant B$ almost surely, then with probability at least $1-\delta$,*

$$\sum_{t=1}^{T} X_t \leqslant \frac{3}{2} \sum_{t=1}^{T} \mathbb{E}_t[X_t] + 4B \log(2\delta^{-1}),$$

and

$$\sum_{t=1}^{T} \mathbb{E}_t[X_t] \leqslant 2\sum_{t=1}^{T} X_t + 8B\log(2\delta^{-1}).$$

Proof. This is a direct consequence of Lemma 3.22. □

We now define/recall some notations. Denote $n_m := \tau_m - \tau_{m-1}$. Fix any epoch $m \in [M]$ and any time step t within epoch m. We have $f^\star = \eta$. For any $f \in \mathcal{F}$, we denote $M_t(f) := Q_t((f(x_t)-y_t)^2-(f^\star(x_t)-y_t)^2)$, and $\widehat{R}_m(f) := \sum_{t=1}^{\tau_{m-1}} Q_t(f(x_t)-y_t)^2$. Recall that we have $Q_t = g_m(x_t)$. We define filtration $\mathfrak{F}_t := \sigma((x_1,y_1),\dots,(x_t,y_t))$,[8] and denote $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot \mid \mathfrak{F}_{t-1}]$. We next present concentration results with respect to a general set of regression function $\mathcal{F}$ with finite pseudo dimension.

Lemma 3.24 (Krishnamurthy et al. (2019))**.** *Consider an infinite set of regression function $\mathcal{F}$. Fix any $\delta \in (0,1)$. For any $\tau, \tau' \in [T]$ such that $\tau < \tau'$, with probability at least $1-\frac{\delta}{2}$, we have*

$$\sum_{t=\tau}^{\tau'} M_t(f) \leqslant \sum_{t=\tau}^{\tau'} \frac{3}{2}\mathbb{E}_t[M_t(f)] + C_\delta,$$

and

$$\sum_{t=\tau}^{\tau'} \mathbb{E}_t[M_t(f)] \leqslant 2\sum_{t=\tau}^{\tau'} M_t(f) + C_\delta,$$

where $C_\delta = C\cdot\left(\mathrm{Pdim}(\mathcal{F})\cdot\log T + \log\left(\frac{\mathrm{Pdim}(\mathcal{F})\cdot T}{\delta}\right)\right)$ with a universal constant $C > 0$.

[8] y_t is not observed (and thus not included in the filtration) when $Q_t = 0$. Note that Q_t is measurable with respect to $\sigma((\mathfrak{F}_{t-1}, x_t))$.

3.7.2.2 Supporting Lemmas for Theorem 3.21

Fix any classifier $\widehat{h} : \mathcal{X} \to \{+1, -1, \perp\}$. For any $x \in \mathcal{X}$, we use the notion

$$\begin{aligned}\mathsf{excess}_\gamma(\widehat{h}; x) &:= \mathbb{P}_{y|x}\big(y \neq \mathsf{sign}(\widehat{h}(x))\big) \cdot \mathbb{1}\big(\widehat{h}(x) \neq \perp\big) + \big(1/2 - \gamma\big) \cdot \mathbb{1}\big(\widehat{h}(x) = \perp\big) \\ &\quad - \mathbb{P}_{y|x}\big(y \neq \mathsf{sign}(h^\star(x))\big) \\ &= \mathbb{1}\big(\widehat{h}(x) \neq \perp\big) \cdot \big(\mathbb{P}_{y|x}\big(y \neq \mathsf{sign}(\widehat{h}(x))\big) - \mathbb{P}_{y|x}\big(y \neq \mathsf{sign}(h^\star(x))\big)\big) \\ &\quad + \mathbb{1}\big(\widehat{h}(x) = \perp\big) \cdot \big(\big(1/2 - \gamma\big) - \mathbb{P}_{y|x}\big(y \neq \mathsf{sign}(h^\star(x))\big)\big) \qquad (3.9)\end{aligned}$$

to represent the excess error of $\widehat{h}$ at point $x \in \mathcal{X}$. Excess error of classifier $\widehat{h}$ can be then written as $\mathsf{excess}_\gamma(\widehat{h}) := \mathsf{err}_\gamma(\widehat{h}) - \mathsf{err}(h^\star) = \mathbb{E}_{x \sim \mathcal{D}_\mathcal{X}}[\mathsf{excess}_\gamma(\widehat{h}; x)]$.

We let $\mathcal{E}$ denote the good event considered in Lemma 3.24, we analyze under this event through out the rest of this section. Most lemmas presented in this section are inspired by results provided Zhu and Nowak (2022b). Our main innovation is an inductive analysis of lemmas that eventually relaxes the requirements for approximation error for Theorem 3.21.

General lemmas. We introduce some general lemmas for Theorem 3.21.

Lemma 3.25. *For any* $m \in [M]$, *we have* $g_m(x) = 1 \implies w(x; \mathcal{F}_m) > \frac{\gamma}{2}$.

Proof. We only need to show that $\mathsf{ucb}(x; \mathcal{F}_m) - \mathsf{lcb}(x; \mathcal{F}_m) \leqslant \frac{\gamma}{2} \implies g_m(x) = 0$. Suppose otherwise $g_m(x) = 1$, which implies that both

$$\begin{aligned}&\frac{1}{2} \in \Big(\mathsf{lcb}(x; \mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x; \mathcal{F}_m) + \frac{\gamma}{4}\Big) \quad \text{and} \\ &\Big[\mathsf{lcb}(x; \mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x; \mathcal{F}_m) + \frac{\gamma}{4}\Big] \nsubseteq \Big[\frac{1}{2} - \gamma, \frac{1}{2} + \gamma\Big]. \qquad (3.10)\end{aligned}$$

If $\frac{1}{2} \in \big(\mathsf{lcb}(x; \mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x; \mathcal{F}_m) + \frac{\gamma}{4}\big)$ and $\mathsf{ucb}(x; \mathcal{F}_m) - \mathsf{lcb}(x; \mathcal{F}_m) \leqslant \frac{\gamma}{2}$, we must have $\mathsf{lcb}(x; \mathcal{F}_m) \geqslant \frac{1}{2} - \frac{3}{4}\gamma$ and $\mathsf{ucb}(x; \mathcal{F}_m) \leqslant \frac{1}{2} + \frac{3}{4}\gamma$, which contradicts with Eq. (3.10). □

Lemma 3.26. *Fix any* $m \in [M]$. *Suppose* $\bar{f} \in \mathcal{F}_m$, *we have* $\mathsf{excess}_\gamma(\widehat{h}_m; x) \leqslant 0$ *if* $g_m(x) = 0$.

Proof. Recall that

$$\begin{aligned}\mathsf{excess}_\gamma(\widehat{h}; x) = \mathbb{1}\big(\widehat{h}(x) \neq \perp\big) \cdot \big(\mathbb{P}_{y|x}\big(y \neq \mathrm{sign}(\widehat{h}(x))\big) - \mathbb{P}_{y|x}\big(y \neq \mathrm{sign}(h^\star(x))\big)\big) \\ + \mathbb{1}\big(\widehat{h}(x) = \perp\big) \cdot \big((1/2 - \gamma) - \mathbb{P}_{y|x}\big(y \neq \mathrm{sign}(h^\star(x))\big)\big).\end{aligned}$$

We now analyze the event $\{g_m(x) = 0\}$ in two cases.

Case 1: $\widehat{h}_m(x) = \perp$**.**

Since $\bar{f}(x) \in [\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)]$ and $\kappa \leqslant \frac{\gamma}{4}$ by assumption, we know that $\eta(x) = f^\star(x) \in [\frac{1}{2} - \gamma, \frac{1}{2} + \gamma]$ and thus $\mathbb{P}_y\big(y \neq \mathrm{sign}(h^\star(x))\big) \geqslant \frac{1}{2} - \gamma$. As a result, we have $\mathsf{excess}_\gamma(\widehat{h}_m; x) \leqslant 0$.

Case 2: $\widehat{h}_m(x) \neq \perp$ **but** $\frac{1}{2} \notin (\mathsf{lcb}(x; \mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x; \mathcal{F}_m) + \frac{\gamma}{4})$**.**

Since $\bar{f}(x) \in [\mathsf{lcb}(x; \mathcal{F}_m), \mathsf{ucb}(x; \mathcal{F}_m)]$ and $\kappa \leqslant \frac{\gamma}{4}$ by assumption, we clearly have $\mathrm{sign}(\widehat{h}_m(x)) = \mathrm{sign}(h^\star(x))$ when $\frac{1}{2} \notin (\mathsf{lcb}(x; \mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x; \mathcal{F}_m) + \frac{\gamma}{4})$. We thus have $\mathsf{excess}_\gamma(\widehat{h}_m; x) \leqslant 0$. □

Inductive lemmas. We prove a set of statements for Theorem 3.21 in an inductive way. Fix any epoch $m \in [M]$, we consider

$$\begin{cases} \widehat{R}_m(\bar{f}) - \widehat{R}_m(f^\star) \leqslant \mathbb{E}_t\Big[Q_t\big(\bar{f}(x_t) - f^\star(x_t)\big)^2\Big] + C_\delta \leqslant \dfrac{3}{2} C_\delta \\ \bar{f} \in \mathcal{F}_m \\ \displaystyle\sum_{t=1}^{\tau_{m-1}} \mathbb{E}_t[M_t(f)] \leqslant 4\beta_m, \forall f \in \mathcal{F}_m \\ \displaystyle\sum_{t=1}^{\tau_{m-1}} \mathbb{E}[Q_t(x_t)(f(x_t) - \bar{f}(x_t))^2] \leqslant 9\beta_m, \forall f \in \mathcal{F}_m \\ \mathcal{F}_m \subseteq \mathcal{F}_{m-1} \end{cases}, \tag{3.11}$$

$$\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x)=1)] \leqslant \frac{144\beta_m}{\tau_{m-1}\gamma^2}\cdot\theta_{\bar{f}}^{\mathrm{val}}\Big(\mathcal{F},\gamma/4,\sqrt{\beta_m/\tau_{m-1}}\Big) \leqslant \frac{144\beta_m}{\tau_{m-1}\gamma^2}\cdot\bar{\theta}, \quad (3.12)$$

and

$$\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x)=1)\cdot w(x;\mathcal{F}_m)] \leqslant \frac{72\beta_m}{\tau_{m-1}\gamma}\cdot\theta_{\bar{f}}^{\mathrm{val}}\Big(\mathcal{F},\gamma/4,\sqrt{\beta_m/\tau_{m-1}}\Big) \leqslant \frac{72\beta_m}{\tau_{m-1}\gamma}\cdot\bar{\theta}. \quad (3.13)$$

Lemma 3.27. *Fix any $\overline{m}=[M]$. When $\overline{m}=1,2$ or when Eq. (3.12) holds true for epochs $m=2,3,\ldots,\overline{m}-1$, then Eq. (3.11) holds true for epoch $m=\overline{m}$.*

Proof. The statements in Eq. (3.11) clearly hold true for $m=\overline{m}=1$ since, by definition, $\mathcal{F}_0=\mathcal{F}$ and $\sum_{t=1}^{0}\ldots=0$. We thus only need to consider the case when $\overline{m}\geqslant 2$. We next prove each of the five statements in Eq. (3.11) for epoch $m=\overline{m}$.

1. In the case when $\overline{m}=2$, from Lemma 3.24, we know that

$$\begin{aligned}\widehat{R}_{\overline{m}}(\bar{f})-\widehat{R}_{\overline{m}}(f^\star) &\leqslant \sum_{t=1}^{\tau_{\overline{m}-1}}\frac{3}{2}\cdot\mathbb{E}_t\Big[Q_t\big(\bar{f}(x_t)-f^\star(x_t)\big)^2\Big]+C_\delta\\ &\leqslant 3+C_\delta \leqslant \frac{3}{2}C_\delta,\end{aligned}$$

   where the second line follows from the fact that $\tau_1=2$ (without loss of generality, we assume $C_\delta\geqslant 6$ here).

   We now focus on the case when $\overline{m}\geqslant 3$. We have

$$\begin{aligned}\widehat{R}_{\overline{m}}(\bar{f})-\widehat{R}_{\overline{m}}(f^\star) &\leqslant \sum_{t=1}^{\tau_{\overline{m}-1}}\frac{3}{2}\cdot\mathbb{E}_t\Big[Q_t\big(\bar{f}(x_t)-f^\star(x_t)\big)^2\Big]+C_\delta\\ &\leqslant \frac{3}{2}\sum_{\check{m}=1}^{\overline{m}-1}n_{\check{m}}\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_{\check{m}}(x)=1)]\cdot\kappa^2+C_\delta\\ &\leqslant \frac{3}{2}\left(2+\sum_{\check{m}=2}^{\overline{m}-1}n_{\check{m}}\frac{144\beta_{\check{m}}\cdot\bar{\theta}}{\tau_{\check{m}-1}\gamma^2}\right)\cdot\kappa^2+C_\delta\end{aligned}$$

$$\begin{aligned}
&\leqslant \left(3 + \frac{144\bar{\theta}}{\gamma^2} \cdot \left(\sum_{\check{m}=2}^{\overline{m}-1} \beta_{\check{m}}\right)\right) \cdot \kappa^2 + C_\delta \\
&\leqslant \left(3 + \frac{432\bar{\theta} \cdot M^2}{\gamma^2} \cdot C_\delta\right) \cdot \kappa^2 + C_\delta \\
&\leqslant \frac{3}{2} C_\delta,
\end{aligned}$$

where the first line follows from Lemma 3.24; the second line follows from the fact that $\|\bar{f} - f^\star\|_\infty \leqslant \kappa$; the third line follows from Eq. (3.12); the forth line follows from $n_{\check{m}} = \tau_{\check{m}-1}$; the fifth line follows from the definition of $\beta_{\check{m}}$; and the last line follows from the choice of κ in Eq. (3.8)

2. Since $\mathbb{E}_t[M_t(f)] = \mathbb{E}_t[Q_t(f(x_t) - f^\star(x_t))^2]$, by Lemma 3.24, we have $\widehat{R}_{\overline{m}}(f^\star) \leqslant \widehat{R}_{\overline{m}}(f) + C_\delta/2$ for any $f \in \mathcal{F}$. Combining this with statement 1 leads to

$$\begin{aligned}
\widehat{R}_{\overline{m}}(\bar{f}) &\leqslant \widehat{R}_{\overline{m}}(f) + 2C_\delta \\
&\leqslant \widehat{R}_{\overline{m}}(f) + \beta_{\overline{m}}
\end{aligned}$$

for any $f \in \mathcal{F}$, where the second line follows from the definition of $\beta_{\overline{m}}$. We thus have $\bar{f} \in \mathcal{F}_{\overline{m}}$ based on the elimination rule.

3. Fix any $f \in \mathcal{F}_{\overline{m}}$. We have

$$\begin{aligned}
\sum_{t=1}^{\tau_{\overline{m}-1}} \mathbb{E}_t[M_t(f)] &\leqslant 2 \sum_{t=1}^{\tau_{\overline{m}-1}} M_t(f) + C_\delta \\
&= 2\widehat{R}_{\overline{m}}(f) - 2\widehat{R}_{\overline{m}}(f^\star) + C_\delta \\
&\leqslant 2\widehat{R}_{\overline{m}}(f) - 2\widehat{R}_{\overline{m}}(\bar{f}) + 4C_\delta \\
&\leqslant 2\widehat{R}_{\overline{m}}(f) - 2\widehat{R}_{\overline{m}}(\widehat{f}_{\overline{m}}) + 4C_\delta \\
&\leqslant 2\beta_{\overline{m}} + 4C_\delta \\
&\leqslant 4\beta_{\overline{m}},
\end{aligned}$$

where the first line follows from Lemma 3.24; the third line follows from

statement 1; the fourth line follows from the fact that $\widehat{f}_{\overline{m}}$ is the minimizer of $\widehat{R}_{\overline{m}}(\cdot)$; and the fifth line follows from the fact that $f \in \mathcal{F}_{\overline{m}}$.

4. Fix any $f \in \mathcal{F}_{\overline{m}}$. We have

$$\begin{aligned}
&\sum_{t=1}^{\tau_{\overline{m}-1}} \mathbb{E}_t[Q_t(x_t)(f(x_t) - \bar{f}(x_t))^2] \\
&= \sum_{t=1}^{\tau_{\overline{m}-1}} \mathbb{E}_t[Q_t(x_t)((f(x_t) - f^\star(x_t)) + (f^\star(x_t) - \bar{f}(x_t)))^2] \\
&\leqslant 2 \sum_{t=1}^{\tau_{\overline{m}-1}} \mathbb{E}_t[Q_t(x_t)(f(x_t) - f^\star(x_t))^2] + 2C_\delta \\
&= 2 \sum_{t=1}^{\tau_{\overline{m}-1}} \mathbb{E}_t[M_t(f)] + 2C_\delta \\
&\leqslant 8\beta_{\overline{m}} + 2C_\delta \\
&\leqslant 9\beta_{\overline{m}},
\end{aligned}$$

where the second line follows from $(a+b)^2 \leqslant 2(a^2+b^2)$ and (the proof of) statement 1 on the second line; and the fourth line follows from statement 3.

5. Fix any $f \in \mathcal{F}_{\overline{m}}$. We have

$$\begin{aligned}
\widehat{R}_{\overline{m}-1}(f) - \widehat{R}_{\overline{m}-1}(\widehat{f}_{\overline{m}-1}) &\leqslant \widehat{R}_{\overline{m}-1}(f) - \widehat{R}_{\overline{m}-1}(f^\star) + \frac{C_\delta}{2} \\
&= \widehat{R}_{\overline{m}}(f) - \widehat{R}_{\overline{m}}(f^\star) - \sum_{t=\tau_{\overline{m}-2}+1}^{\tau_{\overline{m}-1}} M_t(f) + \frac{C_\delta}{2} \\
&\leqslant \widehat{R}_{\overline{m}}(f) - \widehat{R}_{\overline{m}}(\bar{f}) + \frac{3}{2}C_\delta - \sum_{t=\tau_{\overline{m}-2}+1}^{\tau_{\overline{m}-1}} \mathbb{E}_t[M_t(f)]/2 + C_\delta \\
&\leqslant \widehat{R}_{\overline{m}}(f) - \widehat{R}_{\overline{m}}(\widehat{f}_{\overline{m}}) + \frac{5}{2}C_\delta \\
&\leqslant \beta_{\overline{m}} + 3C_\delta \\
&\leqslant \beta_{\overline{m}-1},
\end{aligned}$$

where the first line follows from Lemma 3.24; the third line follows from statement 1 and Lemma 3.24; the fourth line follows from the fact that $\widehat{f}_{\overline{m}}$ is the minimizer with respect to $\widehat{R}_{\overline{m}}$ and Lemma 3.24; the last line follows from the construction of $\beta_{\overline{m}}$.

□

We introduce more notations. Denote $(\mathcal{X}, \Sigma, \mathcal{D}_{\mathcal{X}})$ as the (marginal) probability space, and denote $\overline{\mathcal{X}}_m := \{x \in \mathcal{X} : g_m(x) = 1\} \in \Sigma$ be the region where query *is* requested within epoch m. Under the prerequisites of Lemma 3.28 and Lemma 3.29 (i.e., Eq. (3.11) holds true for epochs $m = 1, 2, \ldots, \overline{m}$), we have $\mathcal{F}_m \subseteq \mathcal{F}_{m-1}$ for $m = 1, 2, \ldots, \overline{m}$, which leads to $\overline{\mathcal{X}}_m \subseteq \overline{\mathcal{X}}_{m-1}$ for $m = 1, 2, \ldots, \overline{m}$. We now define a sub probability measure $\bar{\mu}_m := (\mathcal{D}_{\mathcal{X}})_{|\overline{\mathcal{X}}_m}$ such that $\bar{\mu}_m(\omega) = \mathcal{D}_{\mathcal{X}}(\omega \cap \overline{\mathcal{X}}_m)$ for any $\omega \in \Sigma$. Fix any epoch $m \leqslant \overline{m}$ and consider any measurable function F (that is $\mathcal{D}_{\mathcal{X}}$ integrable), we have

$$\begin{aligned}
\mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}[\mathbb{1}(g_{\overline{m}}(x) = 1) \cdot F(x)] &= \int_{x\in\overline{\mathcal{X}}_{\overline{m}}} F(x)\, d\mathcal{D}_{\mathcal{X}}(x) \\
&\leqslant \int_{x\in\overline{\mathcal{X}}_m} F(x)\, d\mathcal{D}_{\mathcal{X}}(x) \\
&= \int_{x\in\mathcal{X}} F(x)\, d\bar{\mu}_m(x) \\
&=: \mathbb{E}_{x\sim\bar{\mu}_m}[F(x)], \qquad (3.14)
\end{aligned}$$

where, by a slightly abuse of notations, we use $\mathbb{E}_{x\sim\mu}[\cdot]$ to denote the integration with any sub probability measure μ. In particular, Eq. (3.14) holds with equality when $m = \overline{m}$.

Lemma 3.28. *Fix any epoch $\overline{m} \geqslant 2$. Suppose Eq. (3.11) holds true for epochs $m = 1, 2, \ldots, \overline{m}$, we then have Eq. (3.12) holds true for epoch $m = \overline{m}$.*

Proof. We prove Eq. (3.12) for epoch $m = \overline{m}$. We know that $\mathbb{1}(g_{\overline{m}}(x) = 1) = \mathbb{1}(g_{\overline{m}}(x) = 1) \cdot \mathbb{1}(w(x; \mathcal{F}_{\overline{m}}) > \gamma/2)$ from Lemma 3.25. Thus, for any $\check{m} \leqslant \overline{m}$, we

have

$$\begin{aligned}\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_{\overline{m}}(x)=1)] &= \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_{\overline{m}}(x)=1)\cdot\mathbb{1}(w(x;\mathcal{F}_{\overline{m}})>\gamma/2)]\\ &\leqslant \mathbb{E}_{x\sim\bar{\mu}_{\check{m}}}[\mathbb{1}(w(x;\mathcal{F}_{\overline{m}})>\gamma/2)]\\ &\leqslant \mathbb{E}_{x\sim\bar{\mu}_{\check{m}}}\Big(\mathbb{1}\big(\sup_{f\in\mathcal{F}_{\overline{m}}}\big|f(x)-\bar{f}(x)\big|>\gamma/4\big)\Big),\end{aligned} \tag{3.15}$$

where the second line uses Eq. (3.14) and the last line follows from the facts that $\bar{f}\in\mathcal{F}_{\overline{m}}$ (by Eq. (3.11)) and $w(x;\mathcal{F}_{\overline{m}})>\gamma/2 \implies \exists f\in\mathcal{F}_{\overline{m}}, |f(x)-\bar{f}(x)|>\gamma/4$.

For any time step t, let $m(t)$ denote the epoch where t belongs to. From Eq. (3.11), we know that, $\forall f\in\mathcal{F}_{\overline{m}}$,

$$\begin{aligned}9\beta_{\overline{m}} &\geqslant \sum_{t=1}^{\tau_{\overline{m}-1}}\mathbb{E}_t\Big[Q_t\big(f(x_t)-\bar{f}(x_t)\big)^2\Big]\\ &= \sum_{t=1}^{\tau_{\overline{m}-1}}\mathbb{E}_{x\sim\mathcal{D}_X}\Big[\mathbb{1}(g_{m(t)}(x)=1)\cdot\big(f(x)-\bar{f}(x)\big)^2\Big]\\ &= \sum_{\check{m}=1}^{\overline{m}-1}n_{\check{m}}\cdot\mathbb{E}_{x\sim\bar{\mu}_{\check{m}}}\Big[\big(f(x)-\bar{f}(x)\big)^2\Big]\\ &= \tau_{\overline{m}-1}\mathbb{E}_{x\sim\bar{\nu}_{\overline{m}}}\Big[\big(f(x)-\bar{f}(x)\big)^2\Big],\end{aligned} \tag{3.16}$$

where we use $Q_t=g_{m(t)}(x_t)=\mathbb{1}(g_{m(t)}(x)=1)$ and Eq. (3.14) on the second line, and define a new sub probability measure

$$\bar{\nu}_{\overline{m}} := \frac{1}{\tau_{\overline{m}-1}}\sum_{\check{m}=1}^{\overline{m}-1}n_{\check{m}}\cdot\bar{\mu}_{\check{m}}$$

on the third line.

Plugging Eq. (3.16) into Eq. (3.15) leads to the bound

$$\begin{aligned}&\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_{\overline{m}}(x)=1)]\\ &\leqslant \mathbb{E}_{x\sim\bar{\nu}_{\overline{m}}}\Big[\mathbb{1}\Big(\exists f\in\mathcal{F}, \big|f(x)-\bar{f}(x)\big|>\gamma/4, \mathbb{E}_{x\sim\bar{\nu}_{\overline{m}}}\Big[\big(f(x)-\bar{f}(x)\big)^2\Big]\leqslant\frac{9\beta_{\overline{m}}}{\tau_{\overline{m}-1}}\Big)\Big],\end{aligned}$$

where we use the definition of $\bar{\nu}_{\bar{m}}$ again (note that Eq. (3.15) works with any $\check{m} \leqslant \bar{m}$). Based on the Definition 3.20,[9] we then have

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_{\bar{m}}(x)=1)] \\
&\leqslant \frac{144\beta_{\bar{m}}}{\tau_{\bar{m}-1}\gamma^2}\cdot\theta_{\bar{f}}^{\mathrm{val}}\left(\mathcal{F},\gamma/4,\sqrt{9\beta_{\bar{m}}/2\tau_{\bar{m}-1}}\right) \\
&\leqslant \frac{144\beta_{\bar{m}}}{\tau_{\bar{m}-1}\gamma^2}\cdot\theta_{\bar{f}}^{\mathrm{val}}\left(\mathcal{F},\gamma/4,\sqrt{\beta_{\bar{m}}/\tau_{\bar{m}-1}}\right) \\
&\leqslant \frac{144\beta_{\bar{m}}}{\tau_{\bar{m}-1}\gamma^2}\cdot\bar{\theta}.
\end{aligned}$$

□

Lemma 3.29. *Fix any epoch $\bar{m} \geqslant 2$. Suppose Eq. (3.11) holds true for epochs $m = 1, 2, \ldots, \bar{m}$, we then have Eq. (3.13) holds true for epoch $m = \bar{m}$.*

Proof. We prove Eq. (3.13) for epoch $m = \bar{m}$. Similar to the proof of Lemma 3.28, we have

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_{\bar{m}}(x)=1)\cdot w(x;\mathcal{F}_{\bar{m}})] \\
&= \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_{\bar{m}}(x)=1)\cdot\mathbb{1}(w(x;\mathcal{F}_{\bar{m}})>\gamma/2)\cdot w(x;\mathcal{F}_{\bar{m}})] \\
&\leqslant \mathbb{E}_{x\sim\bar{\mu}_{\check{m}}}[\mathbb{1}(w(x;\mathcal{F}_{\bar{m}})>\gamma/2)\cdot w(x;\mathcal{F}_{\bar{m}})]
\end{aligned}$$

for any $\check{m} \leqslant \bar{m}$. With $\bar{\nu}_{\bar{m}} := \frac{1}{\tau_{\bar{m}-1}}\sum_{\check{m}=1}^{\bar{m}-1} n_{\check{m}}\cdot\bar{\mu}_{\check{m}}$, we then have

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_{\bar{m}}(x)=1)\cdot w(x;\mathcal{F}_{\bar{m}})] \\
&\leqslant \mathbb{E}_{x\sim\bar{\nu}_{\bar{m}}}[\mathbb{1}(w(x;\mathcal{F}_{\bar{m}})>\gamma/2)\cdot w(x;\mathcal{F}_{\bar{m}})] \\
&\leqslant \mathbb{E}_{x\sim\bar{\nu}_{\bar{m}}}\left[\mathbb{1}\left(\sup_{f\in\mathcal{F}_{\bar{m}}}\left|f(x)-\bar{f}(x)\right|>\gamma/4\right)\cdot\left(\sup_{f,f'\in\mathcal{F}_{\bar{m}}}|f(x)-f'(x)|\right)\right] \\
&\leqslant 2\mathbb{E}_{x\sim\bar{\nu}_{\bar{m}}}\left[\mathbb{1}\left(\sup_{f\in\mathcal{F}_{\bar{m}}}\left|f(x)-\bar{f}(x)\right|>\gamma/4\right)\cdot\left(\sup_{f\in\mathcal{F}_{\bar{m}}}|f(x)-\bar{f}(x)|\right)\right]
\end{aligned}$$

[9] Note that analyzing with a sub probability measure $\bar{\nu}$ does not cause any problem. See Zhu and Nowak (2022b) for detailed discussion.

$$
\begin{aligned}
&\leqslant 2\int_{\gamma/4}^{1} \mathbb{E}_{x\sim\bar{\nu}_{\overline{m}}}\left[\mathbb{1}\left(\sup_{f\in\mathcal{F}_{\overline{m}}}\left|f(x)-\bar{f}(x)\right|\geqslant\omega\right)\right]\mathrm{d}\,\omega\\
&\leqslant 2\int_{\gamma/4}^{1}\frac{1}{\omega^2}\,\mathrm{d}\,\omega\cdot\left(\frac{9\beta_{\overline{m}}}{\tau_{\overline{m}-1}}\cdot\theta_{\bar{\mathcal{F}}}^{\mathrm{val}}\left(\mathcal{F},\gamma/4,\sqrt{9\beta_{\overline{m}}/2\tau_{\overline{m}-1}}\right)\right)\\
&\leqslant \frac{72\beta_{\overline{m}}}{\tau_{\overline{m}-1}\,\gamma}\cdot\theta_{\bar{\mathcal{F}}}^{\mathrm{val}}\left(\mathcal{F},\gamma/4,\sqrt{\beta_{\overline{m}}/\tau_{\overline{m}-1}}\right)\\
&\leqslant \frac{72\beta_{\overline{m}}}{\tau_{\overline{m}-1}\,\gamma}\cdot\bar{\theta},
\end{aligned}
$$

where we follow similar steps as in the proof of Lemma 3.28 and use some basic arithmetic facts. □

Lemma 3.30. *Eq. (3.11), Eq. (3.12) and Eq. (3.13) hold true for all* $m\in[M]$.

Proof. We first notice that, by Lemma 3.27, Eq. (3.11) holds true for epochs $\overline{m}=1,2$ unconditionally. We also know that, by Lemma 3.28 and Lemma 3.29, once Eq. (3.11) holds true for epochs $m=1,2,\ldots,\overline{m}$, Eq. (3.12) and Eq. (3.13) hold true for epochs $m=\overline{m}$ as well; at the same time, by Lemma 3.27, once Eq. (3.12) holds true for epochs $m=2,3,\ldots,\overline{m}$, Eq. (3.11) will hold true for epoch $m=\overline{m}+1$.

We thus can start the induction procedure from $\overline{m}=2$, and make sure that Eq. (3.11), Eq. (3.12) and Eq. (3.13) hold true for all $m\in[M]$. □

3.7.3 Proof of Theorem 3.21

Theorem 3.21. *Suppose* $\theta_{\mathcal{F}}^{\mathrm{val}}(\gamma/4)\leqslant\bar{\theta}$ *and the approximation level* $\kappa\in(0,\gamma/4]$ *satisfies*

$$
\left(\frac{432\bar{\theta}\cdot M^2}{\gamma^2}\right)\cdot\kappa^2\leqslant\frac{1}{10}. \tag{3.8}
$$

With probability at least $1-\delta$, *Algorithm 7 returns a classifier* $\widehat{h}:\mathcal{X}\to\{+1,-1,\perp\}$ *with Chow's excess error*

$$
\mathsf{excess}_{\gamma}(\widehat{h})=O\left(\varepsilon\cdot\log\left(\frac{\bar{\theta}\cdot\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}\right)\right),
$$

after querying at most

$$O\left(\frac{M^2 \cdot \mathrm{Pdim}(\mathcal{F}) \cdot \log(T/\delta) \cdot \bar{\theta}}{\gamma^2}\right)$$

labels.

Proof. We analyze under the good event $\mathcal{E}$ defined in Lemma 3.24, which holds with probability at least $1 - \frac{\delta}{2}$. Note that all supporting lemmas stated in Section 3.7.2.2 hold true under this event.

Fix any $m \in [M]$. We analyze the Chow's excess error of $\widehat{h}_m$, which is measurable with respect to $\mathfrak{F}_{\tau_{m-1}}$. For any $x \in \mathcal{X}$, if $g_m(x) = 0$, Lemma 3.26 implies that $\mathsf{excess}_\gamma(\widehat{h}_m; x) \leqslant 0$. If $g_m(x) = 1$, we know that $\widehat{h}_m(x) \neq \perp$ and $\frac{1}{2} \in (\mathsf{lcb}(x; \mathcal{F}_m) - \frac{\gamma}{4}, \mathsf{ucb}(x; \mathcal{F}_m) + \frac{\gamma}{4})$. Since $\bar{f} \in \mathcal{F}_m$ by Lemma 3.30 (with Eq. (3.11)) and $\sup_{x \in \mathcal{X}} |\bar{f}(x) - f^\star(x)| \leqslant \kappa \leqslant \gamma/4$ by construction. The error incurred in this case is upper bounded by

$$\begin{aligned}
\mathsf{excess}(\widehat{h}_m; x) &\leqslant 2|f^\star(x) - 1/2| \\
&\leqslant 2\kappa + 2|\bar{f}(x) - 1/2| \\
&\leqslant 2\kappa + 2w(x; \mathcal{F}_m) + \frac{\gamma}{2} \\
&\leqslant 4w(x; \mathcal{F}_m),
\end{aligned}$$

where we use Lemma 3.25 in the last line.

Combining these two cases together, we have

$$\mathsf{excess}(\widehat{h}_m) \leqslant 4\,\mathbb{E}_{x \sim \mathcal{D}_\mathcal{X}}[\mathbb{1}(g_m(x) = 1) \cdot w(x; \mathcal{F}_m)].$$

Take $m = M$ and apply Lemma 3.30 (and Eq. (3.13)) leads to the following guarantee.

$$\mathsf{excess}(\widehat{h}_M) \leqslant \frac{576\beta_M}{\tau_{M-1}\gamma} \cdot \theta_{\bar{f}}^{\mathrm{val}}\left(\mathcal{F}, \gamma/4, \sqrt{\beta_M/\tau_{M-1}}\right)$$

$$\leqslant O\left(\frac{\mathrm{Pdim}(\mathcal{F})\log(T/\delta)}{T\gamma}\cdot\bar{\theta}\right)$$
$$= O\left(\varepsilon\cdot\log\left(\frac{\bar{\theta}\cdot\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma\,\delta}\right)\right),$$

where we use the fact that $T = \frac{\bar{\theta}\cdot\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}$.

We now analyze the label complexity (note that the sampling process of Algorithm 7 stops at time $t = \tau_{M-1}$). Note that $\mathbb{E}[\mathbb{1}(Q_t = 1) \mid \mathfrak{F}_{t-1}] = \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{1}(g_m(x) = 1)]$ for any epoch $m \geqslant 2$ and time step t within epoch m. Combine Lemma 3.30 with Eq. (3.12) (and Lemma 3.30) leads to

$$\begin{aligned}
\sum_{t=1}^{\tau_{M-1}}\mathbb{1}(Q_t = 1) &\leqslant \frac{3}{2}\sum_{t=1}^{\tau_{M-1}}\mathbb{E}[\mathbb{1}(Q_t = 1)\mid\mathfrak{F}_{t-1}] + 4\log(2/\delta)\\
&\leqslant 3 + \frac{3}{2}\sum_{m=2}^{M-1}\frac{(\tau_m - \tau_{m-1})\cdot 144\beta_m}{\tau_{m-1}\,\gamma^2}\cdot\bar{\theta} + 4\log(2/\delta)\\
&\leqslant 3 + 4\log(2/\delta) + O\left(\frac{M^2\cdot\mathrm{Pdim}(\mathcal{F})\cdot\log(T/\delta)\cdot\bar{\theta}}{\gamma^2}\right)\\
&= O\left(\frac{M^2\cdot\mathrm{Pdim}(\mathcal{F})\cdot\log(T/\delta)\cdot\bar{\theta}}{\gamma^2}\right)
\end{aligned}$$

with probability at least $1-\delta$ (due to another application of Lemma 3.23 with confidence level $\delta/2$); where we use the fact that $T = \frac{\bar{\theta}\cdot\mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma}$. □

3.8 Other Proofs and Supporting Results

3.8.1 Proofs and Supporting Results for Section 3.2.2

Proposition 3.6. *Suppose $\mathcal{D}_{XY} \in \mathcal{P}(\alpha,\beta)$. One can construct a set of neural network classifier $\mathcal{H}_{\mathsf{dnn}}$ such that the following two properties hold simultaneously:*

$$\inf_{h\in\mathcal{H}_{\mathsf{dnn}}}\mathrm{err}(h) - \mathrm{err}(h^\star) = O(\varepsilon) \quad \textit{and} \quad \mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}}) = \widetilde{O}(\varepsilon^{-\frac{d}{\alpha(1+\beta)}}).$$

Proof. We take $\kappa = \varepsilon^{\frac{1}{1+\beta}}$ in Theorem 3.4 to construct a set of neural network classifiers $\mathcal{H}_{\mathsf{dnn}}$ with $W = O(\varepsilon^{-\frac{d}{\alpha(1+\beta)}} \log \frac{1}{\varepsilon})$ total parameters arranged in $L = O(\log \frac{1}{\varepsilon})$ layers. According to Theorem 3.5, we know

$$\mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}}) = O(\varepsilon^{-\frac{d}{\alpha(1+\beta)}} \cdot \log^2(\varepsilon^{-1})) = \widetilde{O}(\varepsilon^{-\frac{d}{\alpha(1+\beta)}}).$$

We now show that there exists a classifier $\bar{h} \in \mathcal{H}_{\mathsf{dnn}}$ with small excess error. Let $\bar{h} = h_{\bar{f}}$ be the classifier such that $\|\bar{f} - \eta\|_\infty \leqslant \kappa$. We can see that

$$\begin{aligned}
\mathsf{excess}(\bar{h}) &= \mathbb{E}\big[\mathbb{1}(\bar{h}(x) \neq y) - \mathbb{1}(h^\star(x) \neq y)\big] \\
&= \mathbb{E}\big[|2\eta(x) - 1| \cdot \mathbb{1}(\bar{h}(x) \neq h^\star(x))\big] \\
&\leqslant 2\kappa \cdot \mathbb{P}_{x \sim \mathcal{D}_X}(x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \kappa) \\
&= O(\kappa^{1+\beta}) \\
&= O(\varepsilon),
\end{aligned}$$

where the third line follows from the fact that $\bar{h}$ and $h^\star$ disagrees only within region $\{x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \kappa\}$ and the incurred error is at most 2κ on each disagreed data point. The fourth line follows from the Tsybakov noise condition and the last line follows from the selection of κ. □

Before proving Theorem 3.7, we first recall the excess error guarantee for empirical risk minimization under Tsybakov noise condition.

Theorem 3.31 (Boucheron et al. (2005))**.** *Suppose $\mathcal{D}_{XY}$ satisfies Tsybakov noise condition with parameter $\beta \geqslant 0$. Consider a datatset $D_n = \{(x_i, y_i)\}_{i=1}^n$ of n points i.i.d. sampled from $\mathcal{D}_{XY}$. Let $\widehat{h} \in \mathcal{H}$ be the empirical risk minimizer on D_n. For any constant $\rho > 0$, we have*

$$\begin{aligned}
&\mathrm{err}(\widehat{h}) - \min_{h \in \mathcal{H}} \mathrm{err}(h) \\
&\leqslant \rho \cdot (\min_{h \in \mathcal{H}} \mathrm{err}(h) - \mathrm{err}(h^\star)) + O\left(\frac{(1+\rho)^2}{\rho} \cdot \left(\frac{\mathrm{VCdim}(\mathcal{H}) \cdot \log n}{n}\right)^{\frac{1+\beta}{2+\beta}} + \frac{\log \delta^{-1}}{n}\right),
\end{aligned}$$

with probability at least $1-\delta$.

Theorem 3.7. *Suppose* $\mathcal{D}_{XY} \in \mathcal{P}(\alpha,\beta)$. *Fix any* $\varepsilon,\delta>0$. *Let* $\mathcal{H}_{\mathsf{dnn}}$ *be the set of neural network classifiers constructed in Proposition 3.6. With* $n=\widetilde{O}(\varepsilon^{-\frac{d+2\alpha+\alpha\beta}{\alpha(1+\beta)}})$ *i.i.d. sampled points, with probability at least* $1-\delta$, *the empirical risk minimizer* $\widehat{h}\in\mathcal{H}_{\mathsf{dnn}}$ *achieves excess error* $O(\varepsilon)$.

Proof. Proposition 3.6 certifies $\min_{h\in\mathcal{H}_{\mathsf{dnn}}}\mathrm{err}(h)-\mathrm{err}(h^\star)=O(\varepsilon)$ and $\mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}})=O\left(\varepsilon^{-\frac{d}{\alpha(1+\beta)}}\cdot\log^2(\varepsilon^{-1})\right)$. Take $\rho=1$ in Theorem 3.31, leads to

$$\mathrm{err}(\widehat{h})-\mathrm{err}(h^\star)\leqslant O\left(\varepsilon+\left(\varepsilon^{-\frac{d}{\alpha(1+\beta)}}\cdot\log^2(\varepsilon^{-1})\cdot\frac{\log n}{n}\right)^{\frac{1+\beta}{2+\beta}}+\frac{\log\delta^{-1}}{n}\right),$$

Taking $n=O(\varepsilon^{-\frac{d+2\alpha+\alpha\beta}{\alpha(1+\beta)}}\cdot\log(\varepsilon^{-1})+\varepsilon^{-1}\cdot\log(\delta^{-1}))=\widetilde{O}(\varepsilon^{-\frac{d+2\alpha+\alpha\beta}{\alpha(1+\beta)}})$ thus ensures that $\mathrm{err}(\widehat{h})-\mathrm{err}(h^\star)=O(\varepsilon)$. □

3.8.2 Proofs and Supporting Results for Section 3.2.3

We prove Theorem 3.9 in Section 3.8.2.1 and discuss the disagreement coefficient in Section 3.8.2.2.

3.8.2.1 Proof of Theorem 3.9

Theorem 3.9. *Suppose* $\mathcal{D}_{XY}\in\mathcal{P}(\alpha,\beta)$. *Fix any* $\varepsilon,\delta>0$. *With probability at least* $1-\delta$, *Algorithm 4 returns a classifier* $\widehat{h}\in\mathcal{H}_{\mathsf{dnn}}$ *with excess error* $\widetilde{O}(\varepsilon)$ *after querying* $\widetilde{O}(\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}})\cdot\varepsilon^{-\frac{d+2\alpha}{\alpha+\alpha\beta}})$ *labels.*

Proof. Construct $\mathcal{H}_{\mathsf{dnn}}$ based on Proposition 3.6 such that $\min_{h\in\mathcal{H}_{\mathsf{dnn}}}\mathrm{err}(h)-\mathrm{err}(h^\star)=O(\varepsilon)$ and $\mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}})=\widetilde{O}(\varepsilon^{-\frac{d}{\alpha(1+\beta)}})$. Taking such $\mathcal{H}_{\mathsf{dnn}}$ into Theorem 3.14 leads to the desired result. □

3.8.2.2 Discussion on Disagreement Coefficient in Theorem 3.9

We discuss cases when the (classifier-based) disagreement coefficient with respect to a set of neural networks is well-bounded. As mentioned before, even for simple classifiers such as linear functions, the disagreement coefficient has been analyzed under additional assumptions (Friedman, 2009; Hanneke, 2014). In this section, we analyze the disagreement coefficient for a set of neural networks under additional assumptions on $\mathcal{D}_{\mathcal{X}\mathcal{Y}}$ and $\mathcal{H}_{\mathsf{dnn}}$ (assumptions on $\mathcal{H}_{\mathsf{dnn}}$ can be implemented via proper preprocessing steps). We leave a more comprehensive investigation of the disagreement coefficient for future work.

The first case is when $\mathcal{D}_{\mathcal{X}}$ is supported on countably many data points. The following result show strict improvement over passive learning.

Definition 3.32 (Disagreement core)**.** *For any hypothesis class $\mathcal{H}$ and classifier h, the disagreement core of h with respect to $\mathcal{H}$ under $\mathcal{D}_{\mathcal{X}\mathcal{Y}}$ is defined as*

$$\partial_{\mathcal{H}} h := \lim_{r \to 0} \mathsf{DIS}(\mathcal{B}_{\mathcal{H}}(h, r)). \tag{3.17}$$

Proposition 3.33 (Lemma 7.12 and Theorem 7.14 in Hanneke (2014))**.** *For any hypothesis class $\mathcal{H}$ and classifier h, we have $\theta_h(\varepsilon) = o(1/\varepsilon)$ if and only if $\mathcal{D}_{\mathcal{X}}(\partial_{\mathcal{H}} h) = 0$. In particular, this implies that $\theta_{\mathcal{H}}(\varepsilon) = o(1/\varepsilon)$ whenever $\mathcal{D}_{\mathcal{X}}$ is supported on countably many data points.*

We now discuss conditions under which we can upper bound the disagreement coefficient by $O(1)$, which ensures results in Theorem 3.9 matching the minimax lower bound for active learning, up to logarithmic factors. We introduce the following *decomposable* condition.

Definition 3.34. *A marginal distribution $\mathcal{D}_{\mathcal{X}}$ is ε-decomposable if its (known) support* $\mathrm{supp}(\mathcal{D}_{\mathcal{X}})$ *can be decomposed into connected subsets, i.e.,* $\mathrm{supp}(\mathcal{D}_{\mathcal{X}}) = \cup_{i \in \mathcal{I}} \mathcal{X}_i$*, such that*

$$\mathcal{D}_{\mathcal{X}}(\cup_{i \in \mathcal{I}'} \mathcal{X}_i) = O(\varepsilon),$$

where $\mathcal{I}' := \{i \in \mathcal{I} : \mathcal{D}_{\mathcal{X}}(\mathcal{X}_i) \leqslant \varepsilon\}$.

Remark 3.35. *Note that Definition 3.34 permits a decomposition such that* $|\bar{\mathcal{I}}| = \Omega(\frac{1}{\varepsilon})$ *where* $\bar{\mathcal{I}} = \mathcal{I} \setminus \mathcal{I}'$. *Definition 3.34 requires no knowledge of the index set* $\mathcal{I}$ *or any* $\mathcal{X}_i$*; it also places no restrictions on the conditional probability on each* $\mathcal{X}_i$.

We first give results for a general hypothesis class $\mathcal{H}$ as follows, and then discuss how to bound the disagreement coefficient for a set of neural networks.

Proposition 3.36. *Suppose* $\mathcal{D}_X$ *is decomposable (into* $\cup_{i\in\mathcal{I}}\mathcal{X}_i$*) and the hypothesis class* $\mathcal{H}$ *consists of classifiers whose predication on each* $\mathcal{X}_i$ *is the same, i.e.,* $|\{h(x) : x \in \mathcal{X}_i\}| = 1$ *for any* $h \in \mathcal{H}$ *and* $i \in \mathcal{I}$. *We then have* $\theta_{\mathcal{H}}(\varepsilon) = O(1)$ *for* ε *sufficiently small.*

Proof. Fix any $h \in \mathcal{H}$. we know that for any $h' \in \mathcal{B}_{\mathcal{H}}(h, \varepsilon)$, we must have $\mathsf{DIS}(\{h, h'\}) \subseteq \cup_{i\in\mathcal{I}'}\mathcal{X}_i$ since $\mathcal{D}_X\{x \in \mathcal{X} : h(x) \neq h'(x)\} \leqslant \varepsilon$; and $\{h(x) : x \in \mathcal{X}_i\} = 1$ for any $h \in \mathcal{H}$ and any $\mathcal{X}_i$. This further implies that $\mathbb{P}(\mathsf{DIS}(\mathcal{B}_{\mathcal{H}}(h, \varepsilon)) = O(\varepsilon)$, and thus $\theta_{\mathcal{H}}(\varepsilon) = O(1)$. □

We next discuss conditions under which we can satisfy the prerequisites of Proposition 3.36. Suppose $\mathcal{D}_{XY} \in \mathcal{P}(\alpha, \beta)$. We assume that $\mathcal{D}_X$ is $(\varepsilon^{\frac{\beta}{1+\beta}})$-decomposable, and, for the desired accuracy level ε, we have

$$|\eta(x) - 1/2| \geqslant 2\varepsilon^{\frac{1}{1+\beta}}, \quad \forall x \in \mathrm{supp}(\mathcal{D}_X). \tag{3.18}$$

With the above conditions satisfied, we can filter out neural networks that are clearly not "close" to η. Specifically, with $\kappa = \varepsilon^{\frac{1}{1+\beta}}$ and $\mathcal{F}_{\mathsf{dnn}}$ be the set of neural networks constructed from Proposition 3.6, we consider

$$\widetilde{\mathcal{F}}_{\mathsf{dnn}} := \{f \in \mathcal{F}_{\mathsf{dnn}} : |f(x) - 1/2| \geqslant \varepsilon^{\frac{1}{1+\beta}}, \forall x \in \mathrm{supp}(\mathcal{D}_X)\}, \tag{3.19}$$

which is guaranteed to contain $\bar{f} \in \mathcal{F}_{\mathsf{dnn}}$ such that $\|\bar{f} - \eta\|_\infty \leqslant \varepsilon^{\frac{1}{1+\beta}}$. Now focus on the subset

$$\widetilde{\mathcal{H}}_{\mathsf{dnn}} := \{h_f : f \in \widetilde{\mathcal{F}}_{\mathsf{dnn}}\}. \tag{3.20}$$

We clearly have $h_{\bar{f}} \in \widetilde{\mathcal{H}}_{\mathsf{dnn}}$ (which ensures an $O(\varepsilon)$-optimal classifier) and $\mathrm{VCdim}(\widetilde{\mathcal{H}}_{\mathsf{dnn}}) \leqslant \mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}})$ (since $\widetilde{\mathcal{H}}_{\mathsf{dnn}} \subseteq \mathcal{H}_{\mathsf{dnn}}$). We upper bound the disagreement coefficient $\theta_{\widetilde{\mathcal{H}}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}})$ next.

Proposition 3.37. *Suppose $\mathcal{D}_{XY} \in \mathcal{P}(\alpha, \beta)$ such that $\mathcal{D}_X$ is $(\varepsilon^{\frac{\beta}{1+\beta}})$-decomposable and Eq. (3.18) is satisfied (with the desired accuracy level ε). We then have $\theta_{\widetilde{\mathcal{H}}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) = O(1)$.*

Proof. The proof is similar to the proof of Proposition 3.36. Fix any $h = h_f \in \widetilde{\mathcal{H}}_{\mathsf{dnn}}$. We first argue that, for any $i \in \mathcal{I}$, under Eq. (3.18), $|\{h_f(x) : x \in \mathcal{X}_i\}| = 1$, i.e., for $x \in \mathcal{X}_i$, $h_f(x)$ equals either 1 or 0, but not both: This can be seen from the fact that any $f \in \widetilde{\mathcal{F}}_{\mathsf{dnn}}$ is continuous and satisfies $|f(x) - 1/2| \geqslant \varepsilon^{\frac{1}{1+\beta}}$ for any $x \in \mathcal{X}_i$.

Fix any $h \in \widetilde{\mathcal{H}}_{\mathsf{dnn}}$. We know that for any $h' \in \mathcal{H}_{\widetilde{\mathcal{H}}_{\mathsf{dnn}}}(h, \varepsilon^{\frac{\beta}{1+\beta}})$, we must have $\mathsf{DIS}(\{h, h'\}) \subseteq \cup_{i \in \mathcal{I}'} \mathcal{X}_i$ due to similar reasons argued in the proof of Proposition 3.36. This further implies that $\mathbb{P}(\mathsf{DIS}(\mathcal{B}_{\widetilde{\mathcal{H}}_{\mathsf{dnn}}}(h, \varepsilon^{\frac{\beta}{1+\beta}})) = O(\varepsilon^{\frac{\beta}{1+\beta}})$, and thus $\theta_{\widetilde{\mathcal{H}}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) = O(1)$. □

We next argue that Eq. (3.18) is only needed in an approximate sense. We define the approximate decomposable condition in the following.

Definition 3.38. *A marginal distribution $\mathcal{D}_X$ is (ε, δ)-decomposable if there exists a known subset $\overline{\mathcal{X}} \subseteq \mathrm{supp}(\mathcal{D}_X)$ such that*

$$\mathcal{D}_X(\overline{\mathcal{X}}) \geqslant 1 - \delta, \tag{3.21}$$

and it can be decomposed into connected subsets, i.e., $\overline{\mathcal{X}} = \cup_{i \in \mathcal{I}} \mathcal{X}_i$, such that

$$\mathcal{D}_X(\cup_{i \in \mathcal{I}'} \mathcal{X}_i) = O(\varepsilon),$$

where $\mathcal{I}' := \{i \in \mathcal{I} : \mathcal{D}_X(\mathcal{X}_i) \leqslant \varepsilon\}$.

Suppose $\mathcal{D}_{XY} \in \mathcal{P}(\alpha, \beta)$. We assume that $\mathcal{D}_X$ is $(\varepsilon^{\frac{\beta}{1+\beta}}, \varepsilon^{\frac{\beta}{1+\beta}})$-decomposable (wrt $\overline{\mathcal{X}} \subseteq \mathcal{D}_X$), and, for the desired accuracy level ε, we have

$$|\eta(x) - 1/2| \geqslant 2\varepsilon^{\frac{1}{1+\beta}}, \quad \forall x \in \overline{\mathcal{X}}. \tag{3.22}$$

With the above conditions satisfied, we can filter out neural networks that are clearly not "close" to η. Specifically, with $\kappa = \varepsilon^{\frac{1}{1+\beta}}$ and $\mathcal{F}_{\mathsf{dnn}}$ be the set of neural networks constructed from Proposition 3.6, we consider

$$\overline{\mathcal{F}}_{\mathsf{dnn}} := \{f \in \mathcal{F}_{\mathsf{dnn}} : |f(x) - 1/2| \geqslant \varepsilon^{\frac{1}{1+\beta}}, \forall x \in \overline{\mathcal{X}}\}, \tag{3.23}$$

which is guaranteed to contain $\bar{f} \in \mathcal{F}_{\mathsf{dnn}}$ such that $\|\bar{f} - \eta\|_\infty \leqslant \varepsilon^{\frac{1}{1+\beta}}$. Now focus on the subset

$$\overline{\mathcal{H}}_{\mathsf{dnn}} := \{h_f : f \in \overline{\mathcal{F}}_{\mathsf{dnn}}\}. \tag{3.24}$$

We clearly have $h_{\bar{f}} \in \overline{\mathcal{H}}_{\mathsf{dnn}}$ (which ensures an $O(\varepsilon)$-optimal classifier) and $\mathrm{VCdim}(\overline{\mathcal{H}}_{\mathsf{dnn}}) \leqslant \mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}})$ (since $\overline{\mathcal{H}}_{\mathsf{dnn}} \subseteq \mathcal{H}_{\mathsf{dnn}}$). We upper bound the disagreement coefficient $\theta_{\overline{\mathcal{H}}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}})$ next.

Proposition 3.39. *Suppose $\mathcal{D}_{XY} \in \mathcal{P}(\alpha, \beta)$ such that $\mathcal{D}_X$ is $(\varepsilon^{\frac{1}{1+\beta}}, \varepsilon)$-decomposable (wrt known $\overline{\mathcal{X}} \subseteq \mathrm{supp}(\mathcal{D}_X)$) and Eq. (3.22) is satisfied (with the desired accuracy level ε). We then have $\theta_{\overline{\mathcal{H}}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) = O(1)$.*

Proof. The proof is the same as the proof of Proposition 3.39 except for any $h' \in \mathcal{H}_{\overline{\mathcal{H}}_{\mathsf{dnn}}}(h, \varepsilon^{\frac{\beta}{1+\beta}})$, we must have $\mathsf{DIS}(\{h, h'\}) \subseteq (\cup_{i \in \mathcal{I}'} \mathcal{X}_i) \cup (\mathrm{supp}(\mathcal{D}_X) \setminus \overline{\mathcal{X}})$. Based on the assumption that $\mathcal{D}_X$ is $(\varepsilon^{\frac{1}{1+\beta}}, \varepsilon)$-decomposable, this also leads to $\theta_{\overline{\mathcal{H}}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) = O(1)$. □

3.8.3 Proof of Theorem 3.10

We provide prerequisites in Section 3.8.3.1 and Section 3.8.3.2 and the preprocessing procedures in Section 3.8.3.3. We give the proof of Theorem 3.10 in Section 3.8.3.4.

3.8.3.1 Upper Bound on Pseudo Dimension

We present a result regarding the approximation and an upper bound on the pseudo dimension (i.e., Definition 3.18).

Proposition 3.40. *Suppose $\mathcal{D}_{\mathcal{X}\mathcal{Y}} \in \mathcal{P}(\alpha, \beta)$. One can construct a set of neural network regression functions $\mathcal{F}_{\mathsf{dnn}}$ such that the following two properties hold simultaneously:*

$$\exists f \in \mathcal{F}_{\mathsf{dnn}} \; s.t. \; \|f - f^\star\|_\infty \leqslant \kappa, \quad \textit{and} \quad \mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}}) \leqslant c \cdot \kappa^{-\frac{d}{\alpha}} \log^2(\kappa^{-1}),$$

where $c > 0$ is a universal constant.

Proof. The result follows by combining Theorem 3.4 and Theorem 3.19. □

3.8.3.2 Upper Bounds on Value Function Disagreement Coefficient

We derive upper bounds on the value function disagreement coefficient (i.e., Definition 3.20). We first introduce the (value function) eluder dimension, a complexity measure that is closely related to the value function disagreement coefficient Russo and Van Roy (2013); Foster et al. (2020c).

Definition 3.41 (Value function eluder dimension)**.** *For any $f^\star \in \mathcal{F}$ and $\gamma_0 > 0$, let $\check{\mathfrak{e}}_{f^\star}(\mathcal{F}, \gamma)$ be the length of the longest sequence of data points $x^1, \ldots, x^m$ such that for all i, there exists $f^i \in \mathcal{F}$ such that*

$$|f^i(x^i) - f^\star(x^i)| > \gamma, \quad \textit{and} \quad \sum_{j<i} (f^i(x^j) - f^\star(x^j))^2 \leqslant \gamma^2.$$

The value function eluder dimension is defined as $\mathfrak{e}_{f^\star}(\mathcal{F}, \gamma_0) \coloneqq \sup_{\gamma > \gamma_0} \check{\mathfrak{e}}_{f^\star}(\mathcal{F}, \gamma)$.

The next result shows that the value function disagreement coefficient can be upper bounded by eluder dimension.

Proposition 3.42 (Foster et al. (2020c))**.** *Suppose $\mathcal{F}$ is a uniform Glivenko-Cantelli class. For any $f^\star : \mathcal{X} \to [0, 1]$ and $\gamma, \varepsilon > 0$, we have $\theta^{\mathrm{val}}_{f^\star}(\mathcal{F}, \gamma, \varepsilon) \leqslant 4\, \mathfrak{e}_{f^\star}(\mathcal{F}, \gamma)$.*

We remark here that the requirement that $\mathcal{F}$ is a uniform Glivenko-Cantelli class is rather weak: It is satisfied as long as $\mathcal{F}$ has finite pseudo dimension (Anthony, 2002).

In the following, we only need to derive upper bounds on the value function eluder dimension, which upper bounds on the value function disagreement coefficient.[10] We first define two definitions: (i) the standard definition of covering number (e.g., see Wainwright (2019)), and (ii) a newly-proposed definition of approximate Lipschitzness.

Definition 3.43. *An ι-covering of a set $\mathcal{X}$ with respect to a metric ρ is a set $\{x_1, \ldots, x_N\} \subseteq \mathcal{X}$ such that for each $x \in \mathcal{X}$, there exists some $i \in [N]$ such that $\rho(x, x_i) \leqslant \iota$. The ι-covering number $\mathcal{N}(\iota; \mathcal{X}, \rho)$ is the cardinality of the smallest ι-cover.*

Definition 3.44. *We call a function $f : \mathcal{X} \to \mathbb{R}$ (L, κ)-approximate Lipschitz if*

$$|f(x) - f(x')| \leqslant L \cdot \|x - x'\|_2 + \kappa$$

for any $x, x' \in \mathcal{X}$.

We next provide upper bounds on value function eluder dimension and value function disagreement coefficient.

Theorem 3.45. *Suppose $\mathcal{F}$ is a set of $(L, \kappa/4)$-approximate Lipschitz functions. For any $\kappa' \geqslant \kappa$, we have $\sup_{f \in \mathcal{F}} \mathfrak{e}_f(\mathcal{F}, \kappa') \leqslant 17 \cdot \mathcal{N}(\frac{\kappa'}{8L}; \mathcal{X}, \|\cdot\|_2)$.*

Proof. Fix any $f \in \mathcal{F}$ and $\bar{\kappa} \geqslant \kappa'$. We first give upper bounds on $\check{\mathfrak{e}}_f(\mathcal{F}, \bar{\kappa})$.

We construct $\mathcal{G} := \mathcal{F} - f$, which is a set of $(2L, \kappa/2)$-Lipschitz functions. Fix any eluder sequence $x^1, \ldots, x^m$ at scale $\bar{\kappa}$ and any $\check{x} \in \mathcal{X}$. We claim that $|\{x_j\}_{j \leqslant m} \cap \mathcal{S}| \leqslant 17$ where $\mathcal{S} := \{x \in \mathcal{X} : \|x - \check{x}\|_2 \leqslant \frac{\bar{\kappa}}{8L}\}$. Suppose $\{x_j\}_{j \leqslant m} \cap \mathcal{S} = x_{j_1}, \ldots, x_{j_k}$ (j_i is ordered based on the ordering of $\{x_j\}_{j \leqslant m}$). Since x^{j_k} is added into the eluder sequence, there must exists a $g^{j_k} \in \mathcal{G}$ such that

$$|g^{j_k}(x^{j_k})| > \bar{\kappa}, \quad \text{and} \quad \sum_{j < j_k} (g^{j_k}(x^j))^2 \leqslant \bar{\kappa}^2. \tag{3.25}$$

[10] We focus on Euclidean geometry on $\mathcal{X}$ (i.e., using $\|\cdot\|_2$ norm) in deriving the upper bound. Slightly tighter bounds might be possible with other norms.

Since g^{j_k} is $(2L, \kappa/2)$-Lipschitz, $\bar{\kappa} \geqslant \kappa' \geqslant \kappa$ and $x^{j_k} \in \mathcal{S}$, we must have $g^{j_k}(x) \geqslant \frac{\bar{\kappa}}{4}$ for any $x \in \mathcal{S}$. As a result, we must have $|\{x_j\}_{j<j_k} \cap \mathcal{S}^i| \leqslant 16$ as otherwise the second constraint in Eq. (3.25) will be violated. We cover the space $\mathcal{X}$ with $\mathcal{N}(\frac{\bar{\kappa}}{8L}; \mathcal{X}, \|\cdot\|_2)$ balls of radius $\frac{\bar{\kappa}}{8L}$. Since the eluder sequence contains at most 17 data points within each ball, we know that $\check{\mathfrak{e}}_f(\mathcal{F}, \bar{\kappa}) \leqslant 17 \cdot \mathcal{N}(\frac{\bar{\kappa}}{8L}; \mathcal{X}, \|\cdot\|_2)$.

The desired result follows by noticing that $17 \cdot \mathcal{N}(\frac{\bar{\kappa}}{8L}; \mathcal{X}, \|\cdot\|_2)$ is non-increasing in $\bar{\kappa}$. □

Corollary 3.46. *Suppose $\mathcal{X} \subseteq \mathbb{B}_r^d := \{x \in \mathbb{R}^d : \|x\|_2 \leqslant r\}$ and $\mathcal{F}$ is a set of $(L, \kappa/4)$-approximate Lipschitz functions. For any $\kappa' \geqslant \kappa$, there exists a universal constant $c > 0$, such that $\theta_{\mathcal{F}}^{\mathrm{val}}(\kappa') := \sup_{f \in \mathcal{F}, \iota > 0} \theta_f^{\mathrm{val}}(\mathcal{F}, \kappa', \iota) \leqslant c \cdot (\frac{Lr}{\kappa'})^d$.*

Proof. It is well-known that $\mathcal{N}(\iota; \mathbb{B}_r^d, \|\cdot\|_2) \leqslant (1 + 2r/\iota)^d$ (Wainwright, 2019). The desired result thus follows from combining Theorem 3.45 with Proposition 3.42. □

3.8.3.3 The Preprocessing Step: Clipping and Filtering

Let $\eta : \mathcal{X} \to [0, 1]$ denote the true conditional probability and $\mathcal{F}_{\mathsf{dnn}}$ denote a set of neural network regression functions (e.g., constructed based on Theorem 3.4). We assume that (i) η is L-Lipschitz, and (ii) there exists a $f \in \mathcal{F}$ such that $\|f - \eta\|_\infty \leqslant \kappa$ for some approximation factor $\kappa > 0$. We present the preprocessing step below in Algorithm 8.

Algorithm 8 The Preprocessing Step: Clipping and Filtering

Input: A set of regression functions $\mathcal{F}$, Lipschitz parameter $L > 0$, approximation factor $\kappa > 0$.

1: **Clipping.** Set $\check{\mathcal{F}} := \{\check{f} : f \in \mathcal{F}\}$, where, for any $f \in \mathcal{F}$, we denote

$$\check{f}(x) := \begin{cases} 1, & \text{if } f(x) \geqslant 1; \\ 0, & \text{if } f(x) \leqslant 0; \\ f(x), & \text{o.w.} \end{cases}$$

2: **Filtering.** Set $\widetilde{\mathcal{F}} := \{\check{f} \in \check{\mathcal{F}} : \check{f} \text{ is } (L, 2\kappa)\text{-approximate Lipschitz}\}$

3: **Return** $\widetilde{\mathcal{F}}$.

Proposition 3.47. *Suppose* η *is* L*-Lipschitz and* $\mathcal{F}_{\mathsf{dnn}}$ *is a set of neural networks (of the same architecture) with* W *parameters arranged in* L *layers such that there exists a* $f \in \mathcal{F}_{\mathsf{dnn}}$ *with* $\|f - \eta\|_\infty \leqslant \kappa$*. Let* $\widetilde{\mathcal{F}}_{\mathsf{dnn}}$ *be the set of functions obtained by applying Algorithm 8 on* $\mathcal{F}_{\mathsf{dnn}}$*, we then have (i)* $\mathrm{Pdim}(\widetilde{\mathcal{F}}_{\mathsf{dnn}}) = O(WL\log(W))$*, and (ii) there exists a* $\widetilde{f} \in \widetilde{\mathcal{F}}_{\mathsf{dnn}}$ *such that* $\|\widetilde{f} - \eta\|_\infty \leqslant \kappa$*.*

Proof. Suppose f is a neural network function, we first notice that the "clipping" step can be implemented by adding one additional layer with $O(1)$ additional parameters for each neural network function. More specifically, fix any $f : \mathcal{X} \to \mathbb{R}$, we can set $\check{f}(x) := \mathsf{ReLU}(f(x)) - \mathsf{ReLU}(f(x) - 1)$. Set $\check{\mathcal{F}}_{\mathsf{dnn}} := \{\check{f} : f \in \mathcal{F}_{\mathsf{dnn}}\}$, we then have $\mathrm{Pdim}(\check{\mathcal{F}}_{\mathsf{dnn}}) = O(WL\log(W))$ based on Theorem 3.19. Let $\widetilde{\mathcal{F}}_{\mathsf{dnn}}$ be the filtered version of $\check{\mathcal{F}}_{\mathsf{dnn}}$. Since $\widetilde{\mathcal{F}}_{\mathsf{dnn}} \subseteq \check{\mathcal{F}}_{\mathsf{dnn}}$, we have $\mathrm{Pdim}(\widetilde{\mathcal{F}}_{\mathsf{dnn}}) = O(WL\log(W))$.

Since $\eta : \mathcal{X} \to [0,1]$, we have $\|\check{f} - \eta\|_\infty \leqslant \|f - \eta\|_\infty$, which implies that there must exists a $\check{f} \in \check{\mathcal{F}}_{\mathsf{dnn}}$ such $\|\check{f} - \eta\|_\infty \leqslant \kappa$. To prove the second statement, it suffices to show that the $\check{f} \in \check{\mathcal{F}}$ that achieves κ approximation error is not removed in the "filtering" step, i.e., $\check{f}$ is $(L, 2\kappa)$-approximate Lipschitz. For any $x, x' \in \mathcal{X}$, we have

$$\begin{aligned}|\check{f}(x) - \check{f}(x')| &= |\check{f}(x) - \eta(x) + \eta(x) - \eta(x') + \eta(x') - \check{f}(x')| \\ &\leqslant L\|x - x'\|_2 + 2\kappa,\end{aligned}$$

where we use the L-Lipschitzness of η and the fact that $\|\check{f} - \eta\|_\infty \leqslant \kappa$. □

Proposition 3.48. *Suppose* η *is* L*-Lipschitz and* $\mathcal{X} \subseteq \mathbb{B}_r^d$*. Fix any* $\kappa \in (0, \gamma/32]$*. There exists a set of neural network regression functions* $\mathcal{F}_{\mathsf{dnn}}$ *such that the followings hold simultaneously.*

1. $\mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}}) \leqslant c \cdot \kappa^{-\frac{d}{\alpha}} \log^2(\kappa^{-1})$ *with a universal constant* $c > 0$*.*
2. *There exists a* $\bar{f} \in \mathcal{F}_{\mathsf{dnn}}$ *such that* $\|\bar{f} - \eta\|_\infty \leqslant \kappa$*.*
3. $\theta^{\mathrm{val}}_{\mathcal{F}_{\mathsf{dnn}}}(\gamma/4) := \sup_{f \in \mathcal{F}_{\mathsf{dnn}}, \iota > 0} \theta^{\mathrm{val}}_f(\mathcal{F}_{\mathsf{dnn}}, \gamma/4, \iota) \leqslant c' \cdot (\frac{Lr}{\gamma})^d$ *with a universal constant* $c' > 0$*.*

Proof. Let $\mathcal{F}_{\mathsf{dnn}}$ be obtained by (i) invoking Theorem 3.4 with approximation level κ, and (ii) invoking Algorithm 8 on the set of functions obtained in step (i). The

first two statements follow from Proposition 3.47, and the third statement follows from Corollary 3.46 (note that to achieve guarantees for disagreement coefficient at level $\gamma/4$, we need to have $\kappa \leqslant \gamma/32$ when invoking Theorem 3.4). □

3.8.3.4 Proof of Theorem 3.10

Theorem 3.10. *Fix any $\varepsilon, \delta, \gamma > 0$. With probability at least $1 - \delta$, Algorithm 5 (with an appropriate initialization at line 1) returns a classifier $\widehat{h}$ with Chow's excess error $\widetilde{O}(\varepsilon)$ after querying $\mathrm{poly}(\frac{1}{\gamma}) \cdot \mathrm{polylog}(\frac{1}{\varepsilon\,\delta})$ labels.*

Proof. Let line 1 of Algorithm 5 be the set of neural networks $\mathcal{F}_{\mathsf{dnn}}$ generated from Proposition 3.48 with approximation level $\kappa \in (0, \gamma/32]$ (and constants c, c' specified therein). To apply results derived in Theorem 3.21, we need to satisfying Eq. (3.8), i.e., specifying an approximation level $\kappa \in (0, \gamma/32]$ such that the following holds true

$$\frac{1}{\kappa^2} \geqslant \frac{4320 \cdot c' \cdot (\frac{Lr}{\gamma})^d \cdot \left(\left\lceil \log_2 \left(\frac{c' \cdot (\frac{Lr}{\gamma})^d \cdot c \cdot (\kappa^{-\frac{d}{\alpha}} \log^2(\kappa^{-1}))}{\varepsilon\,\gamma} \right) \right\rceil \right)^2}{\gamma^2}$$

For the setting we considered, i.e., $\mathcal{X} = [0,1]^d$ and $\eta \in \mathcal{W}_1^{\alpha,\infty}(\mathcal{X})$, we have $r = \sqrt{d} = O(1)$ and $L \leqslant \sqrt{d} = O(1)$ (e.g., see Theorem 4.1 in Heinonen (2005)).[11] We thus only need to select a $\kappa \in (0, \gamma/32]$ such that

$$\frac{1}{\kappa} \geqslant \bar{c} \cdot \left(\frac{1}{\gamma}\right)^{\frac{d}{2}+1} \cdot \left(\log \frac{1}{\varepsilon\,\gamma} + \log \frac{1}{\kappa} \right),$$

with a universal constant $\bar{c} > 0$ (that is possibly d-dependent and α-dependent). Since $x \geqslant 2a \log a \implies x \geqslant a \log x$ for any $a > 0$, we can select a $\kappa > 0$ such that

$$\frac{1}{\kappa} = \check{c} \cdot \left(\frac{1}{\gamma}\right)^{\frac{d}{2}+1} \cdot \log \frac{1}{\varepsilon\,\gamma}$$

[11] Recall that we ignore constants that can be potentially α-dependent and d-dependent.

with a universal constant $\check{c} > 0$. With such choice of κ, from Proposition 3.48, we have

$$\mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}}) = O\left(\left(\frac{1}{\gamma}\right)^{\frac{d^2+d}{2\alpha}} \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\,\gamma}\right)\right).$$

Plugging this bound on $\mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}})$ and the upper bound on $\theta^{\mathrm{val}}_{\mathcal{F}_{\mathsf{dnn}}}(\gamma/4)$ from Proposition 3.48 into the guarantee of Theorem 3.21 leads to $\mathrm{excess}_{\gamma}(\widehat{h}) = O(\varepsilon \cdot \log(\frac{1}{\varepsilon\,\gamma\,\delta}))$ after querying

$$O\left(\left(\frac{1}{\gamma}\right)^{d+2+\frac{d^2+d}{2\alpha}} \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\,\gamma\,\delta}\right)\right)$$

labels. □

3.8.4 Proofs and Supporting Results for Section 3.3

We discuss the proper abstention property of classifier learned in Algorithm 5 the computational efficiency of Algorithm 5 in Section 3.8.4.1 and Section 3.8.4.2. We provide the proof of Theorem 3.11 in Section 3.8.4.3.

3.8.4.1 Proper Abstention

We first recall the definition of *proper abstention* proposed in Zhu and Nowak (2022b).

Definition 3.49 (Proper abstention)**.** *A classifier $\widehat{h} : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ enjoys proper abstention if and only if it abstains in regions where abstention is indeed the optimal choice, i.e., $\{x \in \mathcal{X} : \widehat{h}(x) = \perp\} \subseteq \{x \in \mathcal{X} : \eta(x) \in [\frac{1}{2} - \gamma, \frac{1}{2} + \gamma]\} =: \mathcal{X}_\gamma$.*

We next show that the classifier $\widehat{h}$ returned by Algorithm 7 enjoys the proper abstention property. We also convert the abstaining classifier $\widehat{h} : \mathcal{X} \to \mathcal{Y} \cup \{\perp\}$ into a standard classifier $\check{h} : \mathcal{X} \to \mathcal{Y}$ and quantify its standard excess error. The

conversion is through randomizing the prediction of $\widehat{h}$ over its abstention region, i.e., if $\widehat{h}(x) = \perp$, then its randomized version $\check{h}(x)$ predicts $+1/-1$ with equal probability (Puchkin and Zhivotovskiy, 2021).

Proposition 3.50. *The classifier $\widehat{h}$ returned by Algorithm 7 enjoys proper abstention. With randomization over the abstention region, we have the following upper bound on its standard excess error*

$$\operatorname{err}(\check{h}) - \operatorname{err}(h^\star) = \operatorname{err}_\gamma(\widehat{h}) - \operatorname{err}(h^\star) + \gamma \cdot \mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X}_\gamma). \tag{3.26}$$

Proof. The proper abstention property of $\widehat{h}$ returned by Algorithm 7 is achieved via conservation: $\widehat{h}$ will avoid abstention unless it is absolutely sure that abstention is the optimal choice (also see the proof of Lemma 3.26.

Let $\check{h} : \mathcal{X} \to \mathcal{Y}$ be the randomized version of $\bar{h} : \mathcal{X} \to \{+1, -1, \perp\}$ (over the abstention region $\{x \in \mathcal{X} : \widehat{h}(x) = \perp\} \subseteq \mathcal{X}_\gamma$). We can see that, compared to the Chow's abstention error $1/2 - \gamma$, the additional error incurred over the abstention region is exactly $\gamma \cdot \mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X}_\gamma)$. We thus have

$$\operatorname{err}(\widehat{h}) - \operatorname{err}(h^\star) \leqslant \operatorname{err}_\gamma(\widehat{h}) - \operatorname{err}(h^\star) + \gamma \cdot \mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X}_\gamma).$$

□

To characterize the standard excess error of classifier with proper abstention, we only need to upper bound the term $\mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X}_\gamma)$, which does *not* depends on the (random) classifier $\widehat{h}$. Instead, it only depends on the marginal distribution.

We next introduce the Massart (Massart and Nédélec, 2006), which can be viewed as the extreme version of the Tsybakov noise by sending $\beta \to \infty$.

Definition 3.51 (Massart noise)**.** *The marginal distribution $\mathcal{D}_X$ satisfies the Massart noise condition with parameter $\tau_0 > 0$ if $\mathbb{P}(x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \tau_0) = 0$.*

Proposition 3.52. *Suppose Massart noise holds. By setting the abstention parameter $\gamma = \tau_0$ in Algorithm 7 (and randomization over the abstention region), with probability at*

least $1-\delta$, *we obtain a classifier with standard excess error* $\widetilde{O}(\varepsilon)$ *after querying* $\mathrm{poly}(\frac{1}{\tau_0}) \cdot \mathrm{polylog}(\frac{1}{\varepsilon\,\delta})$ *labels.*

Proof. This is a direct consequence of Theorem 3.10 and Proposition 3.50. □

3.8.4.2 Computational Efficiency

We discuss the efficient implementation of Algorithm 7 and its computational complexity in the section. The computational efficiency of Algorithm 7 mainly follows from the analysis in Zhu and Nowak (2022b). We provide the discussion here for completeness.

Regression orcale. We introduce the regression oracle over the set of initialized neural networks $\mathcal{F}_{\mathsf{dnn}}$ (line 1 at Algorithm 5). Given any set $\mathcal{S}$ of weighted examples $(w, x, y) \in \mathbb{R}_+ \times \mathcal{X} \times \mathcal{Y}$ as input, the regression oracle outputs

$$\widehat{f}_{\mathsf{dnn}} := \arg\min_{f \in \mathcal{F}_{\mathsf{dnn}}} \sum_{(w,x,y)\in\mathcal{S}} w(f(x)-y)^2.$$

While the exact computational complexity of such oracle with a set of neural networks remains elusive, in practice, running stochastic gradient descent often leads to great approximations. We quantify the computational complexity in terms of the number of calls to the regression oracle. Any future analysis on such oracle can be incorporated into our guarantees.

We first state some known results in computing the confidence intervals with respect to a general set of regression functions $\mathcal{F}$.

Proposition 3.53 (Krishnamurthy et al. (2017); Foster et al. (2018, 2020c))**.** *Consider the setting studied in Algorithm 7. Fix any epoch* $m \in [M]$ *and denote* $\mathcal{B}_m := \{(x_t, Q_t, y_t)\}_{t=1}^{\tau_{m-1}}$. *Fix any* $\iota > 0$. *For any data point* $x \in \mathcal{X}$, *there exists algorithms* $\mathbf{Alg}_{\mathsf{lcb}}$ *and* $\mathbf{Alg}_{\mathsf{ucb}}$ *that certify*

$$\mathsf{lcb}(x;\mathcal{F}_m) - \iota \leqslant \mathbf{Alg}_{\mathsf{lcb}}(x;\mathcal{B}_m,\beta_m,\iota) \leqslant \mathsf{lcb}(x;\mathcal{F}_m) \quad \textit{and}$$

$$\mathsf{ucb}(x;\mathcal{F}_m) \leqslant \mathbf{Alg}_{\mathsf{ucb}}(x;\mathcal{B}_m,\beta_m,\iota) \leqslant \mathsf{ucb}(x;\mathcal{F}_m)+\iota.$$

The algorithms take $O(\frac{1}{\iota^2}\log\frac{1}{\iota})$ *calls of the regression oracle for general* $\mathcal{F}$ *and take* $O(\log\frac{1}{\iota})$ *calls of the regression oracle if* $\mathcal{F}$ *is convex and closed under pointwise convergence.*

Proof. See Algorithm 2 in Krishnamurthy et al. (2017) for the general case; and Algorithm 3 in Foster et al. (2018) for the case when $\mathcal{F}$ is convex and closed under pointwise convergence. □

We now state the computational guarantee of Algorithm 7, given the regression oracle introduced above.

Theorem 3.54. *Algorithm 7 can be efficiently implemented via the regression oracle and enjoys the same theoretical guarantees stated in Theorem 3.10. The number of oracle calls needed is* $\mathrm{poly}(\frac{1}{\gamma})\cdot\frac{1}{\varepsilon}$*; the per-example inference time of the learned* $\widehat{h}_M$ *is* $\widetilde{O}(\frac{1}{\gamma^2}\cdot\mathrm{polylog}(\frac{1}{\varepsilon\gamma}))$ *for general* $\mathcal{F}$*, and* $\widetilde{O}(\mathrm{polylog}(\frac{1}{\varepsilon\gamma}))$ *when* $\mathcal{F}$ *is convex.*

Proof. Fix any epoch $m \in [M]$. Denote $\bar{\iota} := \frac{\gamma}{8M}$ and $\iota_m := \frac{(M-m)\gamma}{8M}$. With any observed $x \in \mathcal{X}$, we construct the approximated confidence intervals $\widehat{\mathsf{lcb}}(x;\mathcal{F}_m)$ and $\widehat{\mathsf{ucb}}(x;\mathcal{F}_m)$ as follows.

$$\begin{aligned}\widehat{\mathsf{lcb}}(x;\mathcal{F}_m) &:= \mathbf{Alg}_{\mathsf{lcb}}(x;\mathcal{B}_m,\beta_m,\bar{\iota})-\iota_m \quad \text{and}\\ \widehat{\mathsf{ucb}}(x;\mathcal{F}_m) &:= \mathbf{Alg}_{\mathsf{ucb}}(x;\mathcal{B}_m,\beta_m,\bar{\iota})+\iota_m.\end{aligned}$$

For efficient implementation of Algorithm 7, we replace $\mathsf{lcb}(x;\mathcal{F}_m)$ and $\mathsf{ucb}(x;\mathcal{F}_m)$ with $\widehat{\mathsf{lcb}}(x;\mathcal{F}_m)$ and $\widehat{\mathsf{ucb}}(x;\mathcal{F}_m)$ in the construction of $\widehat{h}_m$ and g_m.

Based on Proposition 3.53, we know that

$$\begin{aligned}&\mathsf{lcb}(x;\mathcal{F}_m)-\iota_m-\bar{\iota} \leqslant \widehat{\mathsf{lcb}}(x;\mathcal{F}_m) \leqslant \mathsf{lcb}(x;\mathcal{F}_m)-\iota_m \quad \text{and}\\ &\mathsf{ucb}(x;\mathcal{F}_m)+\iota_m \leqslant \widehat{\mathsf{ucb}}(x;\mathcal{F}_m) \leqslant \mathsf{ucb}(x;\mathcal{F}_m)+\iota_m+\bar{\iota}.\end{aligned}$$

Since $\iota_m+\bar{\iota} \leqslant \frac{\gamma}{8}$ for any $m \in [M]$, the guarantee stated in Lemma 3.25 can be modified as $g_m(x)=1 \implies w(x;\mathcal{F}_m) \geqslant \frac{\gamma}{4}$. The guarantee stated in Lemma 3.26

also holds true since we have $\widehat{\mathsf{lcb}}(x;\mathcal{F}_m) \leqslant \mathsf{lcb}(x;\mathcal{F}_m)$ and $\widehat{\mathsf{ucb}}(x;\mathcal{F}_m) \geqslant \mathsf{ucb}(x;\mathcal{F}_m)$ by construction. Suppose $\mathcal{F}_m \subseteq \mathcal{F}_{m-1}$ (as in Lemma 3.27), we have

$$\widehat{\mathsf{lcb}}(x;\mathcal{F}_m) \geqslant \mathsf{lcb}(x;\mathcal{F}_m) - \iota_m - \bar{\iota} \geqslant \mathsf{lcb}(x;\mathcal{F}_{m-1}) - \iota_{m-1} \geqslant \widehat{\mathsf{lcb}}(x;\mathcal{F}_{m-1}) \quad \text{and}$$
$$\widehat{\mathsf{ucb}}(x;\mathcal{F}_m) \leqslant \mathsf{ucb}(x;\mathcal{F}_m) + \iota_m + \bar{\iota} \leqslant \mathsf{ucb}(x;\mathcal{F}_{m-1}) + \iota_{m-1} \leqslant \widehat{\mathsf{ucb}}(x;\mathcal{F}_{m-1}),$$

which ensures that $\mathbb{1}(g_m(x) = 1) \leqslant \mathbb{1}(g_{m-1}(x) = 1)$. Thus, the inductive lemmas appearing in Section 3.7.2.2 can be proved similarly with changes only in constant terms (also change the constant terms in the definition of $\bar{\theta}$ and in Eq. (3.8), since $\frac{\gamma}{2}$ is replaced by $\frac{\gamma}{4}$ in Lemma 3.25). As a result, the guarantees stated in Theorem 3.21 (and Theorem 3.10) hold true with changes only in constant terms.

We now discuss the computational complexity of the efficient implementation. At the beginning of each epoch m. We use one oracle call to compute $\widehat{f}_m := \arg\min_{f\in\mathcal{F}} \sum_{t=1}^{\tau_{m-1}} Q_t(f(x_t) - y_t)^2$. The main computational cost comes from computing $\widehat{\mathsf{lcb}}$ and $\widehat{\mathsf{ucb}}$ at each time step. We take $\iota = \bar{\iota} := \frac{\gamma}{8M}$ into Proposition 3.53, which leads to $O(\frac{(\log T)^2}{\gamma^2} \cdot \log(\frac{\log T}{\gamma}))$ calls of the regression oracle for general $\mathcal{F}$ and $O(\log(\frac{\log T}{\gamma}))$ calls of the regression oracle for any convex $\mathcal{F}$ that is closed under pointwise convergence. This also serves as the per-example inference time for $\widehat{h}_M$. The total computational cost of Algorithm 7 is derived by multiplying the per-round cost by T and plugging $T = \frac{\theta\, \mathrm{Pdim}(\mathcal{F})}{\varepsilon\,\gamma} = \widetilde{O}(\mathrm{poly}(\frac{1}{\gamma}) \cdot \frac{1}{\varepsilon})$ into the bound . □

3.8.4.3 Proof of Theorem 3.11

For ease of construction, we suppose the instance space is $\mathcal{X} = \mathbb{B}_1^d := \{x \in \mathbb{R}^d : \|x\|_2 \leqslant 1\}$. Part of our construction is inspired by Li et al. (2021).

Theorem 3.11. *Fix any $\gamma \in (0, 1/8)$. For any accuracy level ε sufficiently small, there exists a problem instance such that (1) $\eta \in \mathcal{W}_1^{1,\infty}(\mathcal{X})$ and is of the form $\eta(x) := \mathsf{ReLU}(\langle w, x\rangle + a) + b$; and (2) for any active learning algorithm, it takes at least $\gamma^{-\Omega(d)}$ labels to identify an ε-optimal classifier, for either standard excess error or Chow's excess error (with parameter γ).*

Proof. Fix any $\gamma \in (0, 1/8)$. We first claim that we can find a discrete subset $\overline{\mathcal{X}} \subseteq \mathcal{X}$ with cardinality $|\overline{\mathcal{X}}| \geqslant (1/8\gamma)^{d/2}$ such that $\|x_i\|_2 = 1$ and $\langle x_1, x_2 \rangle \leqslant 1 - 4\gamma$ for any $x_i \in \overline{\mathcal{X}}$. To prove this, we first notice that $\|x_1 - x_2\|_2 \geqslant \tau \iff \langle x_1, x_2 \rangle \leqslant 1 - \tau^2/2$. Since the τ-packing number on the unit sphere is at least $(1/\tau)^d$, setting $\tau = \sqrt{8\gamma}$ leads to the desired claim.

We set $\mathcal{D}_{\mathcal{X}} := \mathrm{unif}(\overline{\mathcal{X}})$ and $\mathcal{F}_{\mathsf{dnn}} := \{\mathsf{ReLU}(\langle w, \cdot \rangle - (1-4\gamma)) + (1/2 - 2\gamma) : w \in \overline{\mathcal{X}}\}$. We have $\mathcal{F}_{\mathsf{dnn}} \subseteq \mathcal{W}_1^{1,\infty}(\mathcal{X})$ since $\|w\|_2 \leqslant$ for any $w \in \overline{\mathcal{X}}$. We randomly select a $w^\star \in \mathcal{X}$ and set $f^\star(\cdot) = \eta(\cdot) = \mathsf{ReLU}(\langle w^\star, \cdot \rangle - (1 - 4\gamma)) + (1/2 - 2\gamma)$. We assume that the labeling feedback is the conditional expectation, i.e., $\eta(x)$ is provided if x is queried. We see that $f^\star(x) = 1/2 - 2\gamma$ for any $x \in \mathcal{X}$ but $x \neq w^\star$, and $f^\star(w^\star) = 1/2 + 2\gamma$. We can see that mistakenly select the wrong $\widehat{f} \neq f^\star$ leads to $\frac{\gamma}{4} \cdot \frac{2}{|\overline{\mathcal{X}}|} = \frac{\gamma}{2|\overline{\mathcal{X}}|}$ excess error. Note that the excess error holds true in both standard excess error and Chow's excess error (with parameter γ) since $\mathcal{D}_{\mathcal{X}}(x \in \mathcal{X} : \eta(x) \in [1/2 - \gamma, 1/2 + \gamma]) = 0$ by construction.

We suppose the desired access error ε is sufficiently small (e.g., $\varepsilon \leqslant \frac{\gamma}{8|\overline{\mathcal{X}}|}$). We now show that, with label complexity at most $K := \lfloor |\overline{\mathcal{X}}|/2 \rfloor = \Omega(\gamma^{-d/2})$, any active learning algorithm will, in expectation, pick a classifier that has $\Omega(\varepsilon)$ excess error. Since the worst case error of any randomized algorithm is lower bounded by the expected error of the best deterministic algorithm against a input distribution (Yao, 1977), we only need to analyze a deterministic learner. We set the input distribution as the uniform distribution over instances with parameter $w^\star \in \overline{\mathcal{X}}$. For any deterministic algorithm, we use $s := (x_{i_1}, \ldots, x_{i_K})$ to denote the data points queried under the constraint that at most K labels can be queried. We denote $\widehat{f} \in \mathcal{F}$ as the learned classifier conditioned on s. Since $w^\star \sim \mathrm{unif}(\overline{\mathcal{X}})$, we know that, with probability at least $\frac{1}{2}$, $w^\star \notin s$. Conditioned on that event, we know that, with probability at least $\frac{1}{2}$, the learner will output $\widehat{f} \neq f^\star$ since more than half of the data points remains unqueried. The deterministic algorithm thus outputs the wrong $\widehat{f} \neq f^\star$ with probability at least $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$, which has $\frac{\gamma}{2|\overline{\mathcal{X}}|}$ excess error as previously discussed. When $\varepsilon \leqslant \frac{\gamma}{8|\overline{\mathcal{X}}|}$, this leads to $\Omega(\varepsilon)$ excess error in expectation. □

3.8.5 Proofs and Supporting Results for Section 3.4

We provide mathematical backgrounds for the Radon BV^2 space in Section 3.8.5.1, derive approximation results and passive learning results in Section 3.8.5.2, and derive active learning results in Section 3.8.5.3.

3.8.5.1 The Radon BV^2 Space

We provide explicit definition of the $\|f\|_{\mathscr{R}\mathsf{BV}^2(\mathcal{X})}$ and associated mathematical backgrounds in this section. Also see Ongie et al. (2020); Parhi and Nowak (2021, 2022b,a); Unser (2022) for more discussions.

We first introduce the *Radon transform* of a function $f : \mathbb{R}^d \to \mathbb{R}$ as

$$\mathscr{R}\{f\}(\gamma, t) := \int_{\{x:\gamma^\top x = t\}} f(x)\, ds(x), \quad (\gamma, t) \in \mathbb{S}^{d-1} \times \mathbb{R},$$

where s denotes the surface measure on the hyperplane $\{x : \gamma^\top x = t\}$. The Radon domain is parameterized by a *direction* $\gamma \in \mathbb{S}^{d-1}$ and an *offset* $t \in \mathbb{R}$. We also introduce the *ramp filter* as

$$\Lambda^{d-1} := (-\partial_t^2)^{\frac{d-1}{2}},$$

where ∂_t denotes the partial derivative with respect to the offset variable, t, of the Radon domain, and the fractional powers are defined in terms of Riesz potentials.

With the above preparations, we can define the $\mathscr{R}\,\mathsf{TV}^2$-seminorm as

$$\mathscr{R}\,\mathsf{TV}^2(f) := c_d \|\partial_t^2 \Lambda^{d-1} \mathscr{R} f\|_{\mathcal{M}(\mathbb{S}^{d-1}\times\mathbb{R})},$$

where $c_d = 1/(2(2\pi)^{d-1})$ is a dimension-dependent constant, and $\|\cdot\|_{\mathcal{M}(\mathbb{S}^{d-1}\times\mathbb{R})}$ denotes the *total variation norm* (in terms of measures) over the bounded domain $\mathbb{S}^{d-1} \times \mathbb{R}$. The $\mathscr{R}\,\mathsf{BV}^2$ norm of f over $\mathbb{R}^d$ is defined as

$$\|f\|_{\mathscr{R}\mathsf{BV}^2(\mathbb{R}^d)} := \mathscr{R}\,\mathsf{TV}^2(f) + |f(0)| + \sum_{k=1}^{d} |f(e_k) - f(0)|,$$

where $\{e_k\}_{k=1}^d$ denotes the canonical basis of $\mathbb{R}^d$. The $\mathscr{R}\,\mathsf{BV}^2(\mathbb{R}^d)$ space is then defined as

$$\mathscr{R}\,\mathsf{BV}^2(\mathbb{R}^d) := \{f \in L^{\infty,1}(\mathbb{R}^d) : \mathscr{R}\,\mathsf{BV}^2(f) < \infty\},$$

where $L^{\infty,1}(\mathbb{R}^d)$ is the Banach space of functions mapping $\mathbb{R}^d \to \mathbb{R}$ of at most linear growth. To define the $\mathscr{R}\,\mathsf{BV}^2$ norm of f over a bounded domain $\mathcal{X} \subseteq \mathbb{R}^d$, we use the standard approach of considering restrictions of functions in $\mathscr{R}\,\mathsf{BV}^2(\mathbb{R}^d)$, i.e.,

$$\|f\|_{\mathscr{R}\,\mathsf{BV}^2(\mathcal{X})} := \inf_{g \in \mathscr{R}\,\mathsf{BV}^2(\mathbb{R}^d)} \|g\|_{\mathscr{R}\,\mathsf{BV}^2(\mathbb{R}^d)} \quad \text{s.t.} \quad g|_{\mathcal{X}} = f.$$

In the rest of Section 3.8.5, we use $\mathcal{P}(\beta)$ to denote the set of distributions that satisfy (1) Tsybakov noise condition with parameter $\beta \geqslant 0$; and (2) $\eta \in \mathscr{R}\,\mathsf{BV}_1^2(\mathcal{X})$.

3.8.5.2 Approximation and Passive Learning Results

Proposition 3.55. *Suppose $\mathcal{D}_{\mathcal{X}\mathcal{Y}} \in \mathcal{P}(\beta)$. One can construct a set of neural network classifier $\mathcal{H}_{\mathsf{dnn}}$ such that the following two properties hold simultaneously:*

$$\min_{h \in \mathcal{H}_{\mathsf{dnn}}} \mathrm{err}(h) - \mathrm{err}(h^\star) = O(\varepsilon) \quad \textit{and} \quad \mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}}) = \widetilde{O}(\varepsilon^{-\frac{2d}{(1+\beta)(d+3)}}).$$

Proof. We take $\kappa = \varepsilon^{\frac{1}{1+\beta}}$ in Theorem 3.12 to construct a set of neural network classifiers $\mathcal{H}_{\mathsf{dnn}}$ with $W = O(\varepsilon^{-\frac{2d}{(1+\beta)(d+3)}})$ total parameters arranged in $L = O(1)$ layers. According to Theorem 3.5, we know

$$\mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}}) = O(\varepsilon^{-\frac{2d}{(1+\beta)(d+3)}} \cdot \log(\varepsilon^{-1})) = \widetilde{O}(\varepsilon^{-\frac{2d}{(1+\beta)(d+3)}}).$$

We now show that there exists a classifier $\bar{h} \in \mathcal{H}_{\mathsf{dnn}}$ with small excess error. Let $\bar{h} = h_{\bar{f}}$ be the classifier such that $\|\bar{f} - \eta\|_\infty \leqslant \kappa$. We can see that

$$\begin{aligned}
\mathsf{excess}(\bar{h}) &= \mathbb{E}\big[\mathbb{1}(\bar{h}(x) \neq y) - \mathbb{1}(h^\star(x) \neq y)\big] \\
&= \mathbb{E}\big[|2\eta(x) - 1| \cdot \mathbb{1}(\bar{h}(x) \neq h^\star(x))\big]
\end{aligned}$$

$$
\begin{aligned}
&\leqslant 2\kappa \cdot \mathbb{P}_{x\sim\mathcal{D}_X}(x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \kappa) \\
&= O(\kappa^{1+\beta}) \\
&= O(\varepsilon),
\end{aligned}
$$

where the third line follows from the fact that $\bar{h}$ and $h^\star$ disagrees only within region $\{x \in \mathcal{X} : |\eta(x) - 1/2| \leqslant \kappa\}$ and the incurred error is at most 2κ on each disagreed data point. The fourth line follows from the Tsybakov noise condition and the last line follows from the selection of κ. □

Theorem 3.56. *Suppose $\mathcal{D}_{XY} \in \mathcal{P}(\beta)$. Fix any $\varepsilon, \delta > 0$. Let $\mathcal{H}_{\mathsf{dnn}}$ be the set of neural network classifiers constructed in Proposition 3.55. With $n = \widetilde{O}(\varepsilon^{-\frac{4d+6+\beta(d+3)}{(1+\beta)(d+3)}})$ i.i.d. sampled data points, with probability at least $1-\delta$, the empirical risk minimizer $\widehat{h} \in \mathcal{H}_{\mathsf{dnn}}$ achieves excess error $O(\varepsilon)$.*

Proof. Proposition 3.55 certifies $\min_{h\in\mathcal{H}_{\mathsf{dnn}}} \mathrm{err}(h) - \mathrm{err}(h^\star) = O(\varepsilon)$ and $\mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}}) = O\left(\varepsilon^{-\frac{2d}{(1+\beta)(d+3)}} \cdot \log(\varepsilon^{-1})\right)$. Take $\rho = 1$ in Theorem 3.31, leads to

$$
\mathrm{err}(\widehat{h}) - \mathrm{err}(h^\star) \leqslant O\left(\varepsilon + \left(\varepsilon^{-\frac{2d}{(1+\beta)(d+3)}} \cdot \log(\varepsilon^{-1}) \cdot \frac{\log n}{n}\right)^{\frac{1+\beta}{2+\beta}} + \frac{\log \delta^{-1}}{n}\right),
$$

Taking $n = O(\varepsilon^{-\frac{4d+6+\beta(d+3)}{(1+\beta)(d+3)}} \cdot \log(\varepsilon^{-1}) + \varepsilon^{-1} \cdot \log(\delta^{-1})) = \widetilde{O}(\varepsilon^{-\frac{4d+6+\beta(d+3)}{(1+\beta)(d+3)}})$ thus ensures that $\mathrm{err}(\widehat{h}) - \mathrm{err}(h^\star) = O(\varepsilon)$. □

3.8.5.3 Active Learning Results

Theorem 3.13. *Suppose $\eta \in \mathscr{R}\,\mathsf{BV}_1^2(\mathcal{X})$ and the Tsybakov noise condition is satisfied with parameter $\beta \geqslant 0$. Fix any $\varepsilon, \delta > 0$. There exists an algorithm such that, with probability at least $1-\delta$, it learns a classifier $\widehat{h} \in \mathcal{H}_{\mathsf{dnn}}$ with excess error $\widetilde{O}(\varepsilon)$ after querying $\widetilde{O}(\theta_{\mathcal{H}_{\mathsf{dnn}}}(\varepsilon^{\frac{\beta}{1+\beta}}) \cdot \varepsilon^{-\frac{4d+6}{(1+\beta)(d+3)}})$ labels.*

Proof. Construct $\mathcal{H}_{\mathsf{dnn}}$ based on Proposition 3.55 such that $\min_{h\in\mathcal{H}_{\mathsf{dnn}}} \mathrm{err}(h) - \mathrm{err}(h^\star) = O(\varepsilon)$ and $\mathrm{VCdim}(\mathcal{H}_{\mathsf{dnn}}) = \widetilde{O}(\varepsilon^{-\frac{2d}{(1+\beta)(d+3)}})$. Taking such $\mathcal{H}_{\mathsf{dnn}}$ as the initialization of Algorithm 6 (line 1) and applying Theorem 3.14 leads to the desired result. □

To derive deep active learning guarantee with abstention in the Radon BV^2 space, we first present two supporting results below.

Proposition 3.57. *Suppose $\mathcal{D}_{\mathcal{X}\mathcal{Y}} \in \mathcal{P}(\beta)$. One can construct a set of neural network regression functions $\mathcal{F}_{\mathsf{dnn}}$ such that the following two properties hold simultaneously:*

$$\exists f \in \mathcal{F}_{\mathsf{dnn}} \text{ s.t. } \|f - f^\star\|_\infty \leqslant \kappa, \quad \text{and} \quad \mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}}) \leqslant c \cdot \kappa^{-\frac{2d}{d+3}} \log^2(\kappa^{-1}),$$

where $c > 0$ is a universal constant.

Proof. The result follows by combining Theorem 3.12 and Theorem 3.19. □

Proposition 3.58. *Suppose η is L-Lipschitz and $\mathcal{X} \subseteq \mathbb{B}^d_r$. Fix any $\kappa \in (0, \gamma/32]$. There exists a set of neural network regression functions $\mathcal{F}_{\mathsf{dnn}}$ such that the followings hold simultaneously.*

1. $\mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}}) \leqslant c \cdot \kappa^{-\frac{2d}{d+3}} \log^2(\kappa^{-1})$ *with a universal constant $c > 0$.*
2. *There exists a $\bar{f} \in \mathcal{F}_{\mathsf{dnn}}$ such that $\|\bar{f} - \eta\|_\infty \leqslant \kappa$.*
3. $\theta^{\mathrm{val}}_{\mathcal{F}_{\mathsf{dnn}}}(\gamma/4) := \sup_{f \in \mathcal{F}_{\mathsf{dnn}}, \iota > 0} \theta^{\mathrm{val}}_f(\mathcal{F}_{\mathsf{dnn}}, \gamma/4, \iota) \leqslant c' \cdot (\frac{Lr}{\gamma})^d$ *with a universal constant $c' > 0$.*

Proof. The implementation and proof are similar to those in Proposition 3.48, except we use Proposition 3.57 instead of Proposition 3.40. □

We now state and prove deep active learning guarantees in the Radon BV^2 space.

Theorem 3.59. *Suppose $\eta \in \mathscr{R}\,\mathsf{BV}^2_1(\mathcal{X})$. Fix any $\varepsilon, \delta, \gamma > 0$. There exists an algorithm such that, with probability at least $1 - \delta$, it learns a classifier $\widehat{h}$ with Chow's excess error $\widetilde{O}(\varepsilon)$ after querying $\mathrm{poly}(\frac{1}{\gamma}) \cdot \mathrm{polylog}(\frac{1}{\varepsilon\,\delta})$ labels.*

Proof. The result is obtained by applying Algorithm 7 with line 1 be the set of neural networks $\mathcal{F}_{\mathsf{dnn}}$ generated from Proposition 3.58 with approximation level $\kappa \in (0, \gamma/32]$ (and constants c, c' specified therein). The rest of the proof proceeds

in a similar way as the proof Theorem 3.10. Since we have $r = 1$ and $L \leqslant 1$ (Parhi and Nowak, 2022a), we only need to choose a $\kappa > 0$ such that

$$\frac{1}{\kappa} = \check{c} \cdot \left(\frac{1}{\gamma}\right)^{\frac{d}{2}+1} \cdot \log \frac{1}{\varepsilon\,\gamma}$$

with a universal constant $\check{c} > 0$. With such choice of κ, we have

$$\mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}}) = O\left(\left(\frac{1}{\gamma}\right)^{\frac{d^2+2d}{d+3}} \mathrm{polylog}\left(\frac{1}{\varepsilon\,\gamma}\right)\right).$$

Plugging this bound on $\mathrm{Pdim}(\mathcal{F}_{\mathsf{dnn}})$ and the upper bound on $\theta^{\mathrm{val}}_{\mathcal{F}_{\mathsf{dnn}}}(\gamma/4)$ from Proposition 3.58 into the guarantee of Theorem 3.21 leads to $\mathrm{excess}_\gamma(\widehat{h}) = O(\varepsilon \cdot \log(\frac{1}{\varepsilon\,\gamma\,\delta}))$ after querying

$$O\left(\left(\frac{1}{\gamma}\right)^{d+2+\frac{d^2+2d}{d+3}} \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\,\gamma\,\delta}\right)\right)$$

labels. □

Part II

Decision Making with Large Action Spaces

4 CONTEXTUAL BANDITS WITH LARGE ACTION SPACES: MADE PRACTICAL

A central problem in sequential decision making is to develop algorithms that are practical and computationally efficient, yet support the use of flexible, general-purpose models. Focusing on the contextual bandit problem, recent progress provides provably efficient algorithms with strong empirical performance when the number of possible alternatives ("actions") is small, but guarantees for decision making in large, continuous action spaces have remained elusive, leading to a significant gap between theory and practice. We present the first efficient, general-purpose algorithm for contextual bandits with continuous, linearly structured action spaces. Our algorithm makes use of computational oracles for (i) supervised learning, and (ii) optimization over the action space, and achieves sample complexity, runtime, and memory independent of the size of the action space. In addition, it is simple and practical. We perform a large-scale empirical evaluation, and show that our approach typically enjoys superior performance and efficiency compared to standard baselines.

4.1 Introduction

We consider the design of practical, theoretically motivated algorithms for sequential decision making with contextual information, better known as the *contextual bandit problem*. Here, a learning agent repeatedly receives a *context* (e.g., a user's profile), selects an *action* (e.g., a news article to display), and receives a *reward* (e.g., whether the article was clicked). Contextual bandits are a useful model for decision making in unknown environments in which both exploration and generalization are required, but pose significant algorithm design challenges beyond classical supervised learning. Recent years have seen development on two fronts: On the theoretical side, extensive research into finite-action contextual bandits has resulted in practical, provably efficient algorithms capable of supporting flexible,

general-purpose models (Langford and Zhang, 2007; Agarwal et al., 2014; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021; Foster and Krishnamurthy, 2021). Empirically, contextual bandits have been widely deployed in practice for online personalization and recommendation tasks (Li et al., 2010; Agarwal et al., 2016; Tewari and Murphy, 2017; Cai et al., 2021), leveraging the availability of high-quality action slates (e.g., subsets of candidate articles selected by an editor).

The developments above critically rely on the existence of a small number of possible decisions or alternatives. However, many applications demand the ability to make contextual decisions in large, potentially continuous spaces, where actions might correspond to images in a database or high-dimensional embeddings of rich documents such as webpages. Contextual bandits in large (e.g., million-action) settings remains a major challenge—both statistically and computationally—and constitutes a substantial gap between theory and practice. In particular:

- Existing *general-purpose* algorithms (Langford and Zhang, 2007; Agarwal et al., 2014; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021; Foster and Krishnamurthy, 2021) allow for the use of flexible models (e.g., neural networks, forests, or kernels) to facilitate generalization across contexts, but have sample complexity and computational requirements linear in the number of actions. These approaches can degrade in performance under benign operations such as duplicating actions.

- While certain recent approaches extend the general-purpose methods above to accommodate large action spaces, they either require sample complexity exponential in action dimension (Krishnamurthy et al., 2020), or require additional distributional assumptions (Sen et al., 2021).

- Various results efficiently handle large or continuous action spaces (Dani et al., 2008; Jun et al., 2017; Yang et al., 2021) with specific types of function approximation, but do not accommodate general-purpose models.

As a result of these algorithmic limitations, empirical aspects of contextual decision making in large action spaces have remained relatively unexplored compared to the

small-action regime (Bietti et al., 2021), with little in the way of readily deployable out-of-the-box solutions.

Contributions. We provide the first efficient algorithms for contextual bandits with continuous, linearly structured action spaces and general function approximation. Following Chernozhukov et al. (2019); Xu and Zeevi (2020); Foster et al. (2020a), we adopt a modeling approach, and assume rewards for each context-action pair (x, a) are structured as

$$f^\star(x, a) = \langle \phi(x, a), g^\star(x) \rangle. \tag{4.1}$$

Here $\phi(x, a) \in \mathbb{R}^d$ is a known context-action embedding (or feature map) and $g^\star \in \mathcal{G}$ is a context embedding to be learned online, which belongs to an arbitrary, user-specified function class $\mathcal{G}$. Our algorithm, SpannerIGW, is computationally efficient (in particular, the runtime and memory are *independent* of the number of actions) whenever the user has access to (i) an *online regression oracle* for supervised learning over the reward function class, and (ii) an *action optimization oracle* capable of solving problems of the form

$$\arg\max_{a \in \mathcal{A}} \langle \phi(x, a), \theta \rangle$$

for any $\theta \in \mathbb{R}^d$. The former oracle follows prior approaches to finite-action contextual bandits (Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021; Foster and Krishnamurthy, 2021), while the latter generalizes efficient approaches to (non-contextual) linear bandits (McMahan and Blum, 2004; Dani et al., 2008; Bubeck et al., 2012; Hazan and Karnin, 2016). We provide a regret bound for SpannerIGW which scales as $\sqrt{\mathrm{poly}(d) \cdot T}$, and—like the computational complexity—is independent of the number of actions. Beyond these results, we provide a particularly practical variant of SpannerIGW (SpannerGreedy), which enjoys even faster runtime at the cost of slightly worse ($\mathrm{poly}(d) \cdot T^{2/3}$-type) regret.

Our techniques. On the technical side, we show how to *efficiently* combine the inverse gap weighting technique (Abe and Long, 1999; Foster and Rakhlin, 2020) previously used in the finite-action setting with optimal design-based approaches for exploration with linearly structured actions. This offers a computational improvement upon the results of Xu and Zeevi (2020); Foster et al. (2020a), which provide algorithms with $\sqrt{\mathrm{poly}(d) \cdot T}$-regret for the setting we consider, but require enumeration over the action space. Conceptually, our results expand upon the class of problems for which minimax approaches to exploration (Foster et al., 2021b) can be made efficient.

Empirical performance. As with previous approaches based on regression oracles, SpannerIGW is simple, practical, and well-suited to flexible, general-purpose function approximation. In extensive experiments ranging from thousands to millions of actions, we find that our methods typically enjoy superior performance compared to existing baselines. In addition, our experiments validate the statistical model in Eq. (4.1) which we find to be well-suited to learning with large-scale language models (Devlin et al., 2019).

4.1.1 Organization

This chapter is organized as follows. In Section 4.2, we formally introduce our statistical model and the computational oracles upon which our algorithms are built; we also discuss additional related work in Section 4.2.2. Subsequent sections are dedicated to our main results.

- As a warm-up, Section 4.3 presents a simplified algorithm, SpannerGreedy, which illustrates the principle of exploration over an approximate optimal design. This algorithm is practical and oracle-efficient, but has suboptimal $\mathrm{poly}(d) \cdot T^{2/3}$-type regret.
- Building on these ideas, Section 4.4 presents our main algorithm, SpannerIGW, which combines the idea of approximate optimal design used by SpannerGreedy

with the inverse gap weighting method (Abe and Long, 1999; Foster and Rakhlin, 2020), resulting in an oracle-efficient algorithm with $\sqrt{\mathrm{poly}(d) \cdot T}$-regret.

Section 4.5 presents empirical results for both algorithms. We close with discussion of future directions in Section 4.6. All proofs are deferred to Section 4.7.

4.2 Problem Setting

The contextual bandit problem proceeds over T rounds. At each round $t \in [T]$, the learner receives a context $x_t \in \mathcal{X}$ (the *context space*), selects an action $a_t \in \mathcal{A}$ (the *action space*), and then observes a reward $r_t(a_t)$, where $r_t : \mathcal{A} \to [-1, 1]$ is the underlying reward function. We assume that for each round t, conditioned on x_t, the reward r_t is sampled from a (unknown) distribution $\mathbb{P}_{r_t}(\cdot \mid x_t)$. We allow both the contexts $x_1, \ldots, x_T$ and the distributions $\mathbb{P}_{r_1}, \ldots, \mathbb{P}_{r_T}$ to be selected in an arbitrary, potentially adaptive fashion based on the history.

Function approximation. Following a standard approach to developing efficient contextual bandit methods, we take a modeling approach, and work with a user-specified class of regression functions $\mathcal{F} \subseteq (\mathcal{X} \times \mathcal{A} \to [-1, 1])$ that aims to model the underlying mean reward function. We make the following realizability assumption (Agarwal et al., 2012; Foster et al., 2018; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021).

Assumption 4.1 (Realizability)**.** *There exists a regression function* $f^\star \in \mathcal{F}$ *such that* $\mathbb{E}[r_t(a) \mid x_t = x] = f^\star(x, a)$ *for all* $a \in \mathcal{A}$ *and* $t \in [T]$.

Without further assumptions, there exist function classes $\mathcal{F}$ for which the regret of any algorithm must grow proportionally to $|\mathcal{A}|$ (e.g., Agarwal et al. (2012)). In order to facilitate generalization across actions and achieve sample complexity and computational complexity independent of $|\mathcal{A}|$, we assume that each function $f \in \mathcal{F}$ is linear in a known (context-dependent) feature embedding of the action.

Following Xu and Zeevi (2020); Foster et al. (2020a), we assume that $\mathcal{F}$ takes the form

$$\mathcal{F} = \{f_g(x, a) = \langle \phi(x, a), g(x) \rangle : g \in \mathcal{G}\},$$

where $\phi(x, a) \in \mathbb{R}^d$ is a known, context-dependent action embedding and $\mathcal{G}$ is a user-specified class of context embedding functions.

This formulation assumes linearity in the action space (after featurization), but allows for nonlinear, *learned* dependence on the context x through the function class $\mathcal{G}$, which can be taken to consist of neural networks, forests, or any other flexible function class a user chooses. For example, in news article recommendation, $\phi(x, a) = \phi(a)$ might correspond to an embedding of an article a obtained using a large pre-trained language-model, while $g(x)$ might correspond to a task-dependent embedding of a user x, which our methods can learn online. Well-studied special cases include the linear contextual bandit setting (Chu et al., 2011; Abbasi-Yadkori et al., 2011), which corresponds to the special case where each $g \in \mathcal{G}$ has the form $g(x) = \theta$ for some fixed $\theta \in \mathbb{R}^d$, as well as the standard finite-action contextual bandit setting, where $d = |\mathcal{A}|$ and $\phi(x, a) = e_a$.

We let $g^\star \in \mathcal{G}$ denote the embedding for which $f^\star = f_{g^\star}$. We assume that $\sup_{x \in \mathcal{X}, a \in \mathcal{A}} \|\phi(x, a)\| \leqslant 1$ and $\sup_{g \in \mathcal{G}, x \in \mathcal{X}} \|g(x)\| \leqslant 1$. In addition, we assume that $\operatorname{span}(\{\phi(x, a)\}) = \mathbb{R}^d$ for all $x \in \mathcal{X}$.

Regret. For each regression function $f \in \mathcal{F}$, let $\pi_f(x_t) := \arg\max_{a \in \mathcal{A}} f(x_t, a)$ denote the induced policy, and define $\pi^\star := \pi_{f^\star}$ as the optimal policy. We measure the performance of the learner in terms of regret:

$$\mathbf{Reg}_{\mathsf{CB}}(T) := \sum_{t=1}^{T} r_t(\pi^\star(x_t)) - r_t(a_t).$$

4.2.1 Computational Oracles

To derive efficient algorithms with sublinear runtime, we make use of two computational oracles: First, following Foster and Rakhlin (2020); Simchi-Levi and Xu (2021); Foster et al. (2020a, 2021a), we use an *online regression oracle* for supervised learning over the reward function class $\mathcal{F}$. Second, we use an *action optimization oracle*, which facilitates linear optimization over the action space $\mathcal{A}$ (McMahan and Blum, 2004; Dani et al., 2008; Bubeck et al., 2012; Hazan and Karnin, 2016).

Function approximation: Regression oracles. A fruitful approach to designing efficient contextual bandit algorithms is through reduction to supervised regression with the class $\mathcal{F}$, which facilitates the use of off-the-shelf supervised learning algorithms and models (Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021; Foster et al., 2020a, 2021a). Following Foster and Rakhlin (2020), we assume access to an *online regression oracle* $\mathbf{Alg}_{\mathsf{Sq}}$, which is an algorithm for online learning (or, sequential prediction) with the square loss.

We consider the following protocol. At each round $t \in [T]$, the oracle produces an estimator $\widehat{f}_t = f_{\widehat{g}_t}$, then receives a context-action-reward tuple $(x_t, a_t, r_t(a_t))$. The goal of the oracle is to accurately predict the reward as a function of the context and action, and we evaluate its prediction error via the square loss $(\widehat{f}_t(x_t, a_t) - r_t)^2$. We measure the oracle's cumulative performance through square-loss regret to $\mathcal{F}$.

Assumption 4.2 (Bounded square-loss regret)**.** *The regression oracle* $\mathbf{Alg}_{\mathsf{Sq}}$ *guarantees that for any* (*potentially adaptively chosen*) *sequence* $\{(x_t, a_t, r_t(a_t))\}_{t=1}^T$,

$$\sum_{t=1}^{T}\left(\widehat{f}_t(x_t, a_t) - r_t(a_t)\right)^2 - \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}(f(x_t, a_t) - r_t(a_t))^2 \leqslant \mathbf{Reg}_{\mathsf{Sq}}(T),$$

for some (*non-data-dependent*) *function* $\mathbf{Reg}_{\mathsf{Sq}}(T)$.

We let $\mathcal{T}_{\mathsf{Sq}}$ denote an upper bound on the time required to (i) query the oracle's estimator $\widehat{g}_t$ with x_t and receive the vector $\widehat{g}_t(x_t) \in \mathbb{R}^d$, and (ii) update the oracle

with the example $(x_t, a_t, r_t(a_t))$. We let $\mathcal{M}_{\mathsf{Sq}}$ denote the maximum memory used by the oracle throughout its execution.

Online regression is a well-studied problem, with computationally efficient algorithms for many models. Basic examples include finite classes $\mathcal{F}$, where one can attain $\mathbf{Reg}_{\mathsf{Sq}}(T) = O(\log|\mathcal{F}|)$ (Vovk, 1998), and linear models $(g(x) = \theta)$, where the online Newton step algorithm (Hazan et al., 2007) satisfies Assumption 4.2 with $\mathbf{Reg}_{\mathsf{Sq}}(T) = O(d \log T)$. More generally, even for classes such as deep neural networks for which provable guarantees may not be available, regression is well-suited to gradient-based methods. We refer to Foster and Rakhlin (2020); Foster et al. (2020a) for more comprehensive discussion.

Large action spaces: Action optimization oracles. The regression oracle setup in the prequel is identical to that considered in the finite-action setting (Foster and Rakhlin, 2020). In order to develop efficient algorithms for large or infinite action spaces, we assume access to an oracle for linear optimization over actions.

Definition 4.3 (Action optimization oracle)**.** *An action optimization oracle* $\mathbf{Alg}_{\mathsf{Opt}}$ *takes as input a context* $x \in \mathcal{X}$*, and vector* $\theta \in \mathbb{R}^d$ *and returns*

$$a^\star := \arg\max_{a \in \mathcal{A}} \langle \phi(x, a), \theta \rangle. \tag{4.2}$$

For a single query to the oracle, We let $\mathcal{T}_{\mathsf{Opt}}$ denote a bound on the runtime for a single query to the oracle. We let $\mathcal{M}_{\mathsf{Opt}}$ denote the maximum memory used by the oracle throughout its execution.

The action optimization oracle in Eq. (4.2) is widely used throughout the literature on linear bandits (Dani et al., 2008; Chen et al., 2017; Cao and Krishnamurthy, 2019; Katz-Samuels et al., 2020), and can be implemented in polynomial time for standard combinatorial action spaces. It is a basic computational primitive in the theory of convex optimization, and when $\mathcal{A}$ is convex, it is equivalent (up to polynomial-time reductions) to other standard primitives such as separation oracles and membership oracles (Schrijver, 1998; Grötschel et al., 2012). It also equivalent

to the well-known Maximum Inner Product Search (MIPS) problem (Shrivastava and Li, 2014), for which sublinear-time hashing based methods are available.

Example 4.4. *Let* $G = (V, E)$ *be a graph, and let* $\phi(x, a) \in \{0, 1\}^{|E|}$ *represent a matching and* $\theta \in \mathbb{R}^{|E|}$ *be a vector of edge weights. The problem of finding the maximum-weight matching for a given set of edge weights can be written as a linear optimization problem of the form in Eq. (4.2), and Edmonds' algorithm (Edmonds, 1965) can be used to find the maximum-weight matching in* $O(|V|^2 \cdot |E|)$ *time.*

Other combinatorial problems that admit polynomial-time action optimization oracles include the maximum-weight spanning tree problem, the assignment problem, and others (Awerbuch and Kleinberg, 2008; Cesa-Bianchi and Lugosi, 2012).

Action representation. We define $b_{\mathcal{A}}$ as the number of bits used to represent actions in $\mathcal{A}$, which is always upper bounded by $O(\log|\mathcal{A}|)$ for finite action sets, and by $\widetilde{O}(d)$ for actions that can be represented as vectors in $\mathbb{R}^d$. Tighter bounds are possible with additional structual assumptions. Since representing actions is a minimal assumption, we hide the dependence on $b_{\mathcal{A}}$ in big-O notation for our runtime and memory analysis.

4.2.2 Additional Related Work

In this section we highlight some relevant lines of research not already discussed.

Efficient general-purpose contextual bandit algorithms. There is a long line of research on computationally efficient methods for contextual bandits with general function approximation, typically based on reduction to either cost-sensitive classification oracles (Langford and Zhang, 2007; Dudik et al., 2011; Agarwal et al., 2014) or regression oracles (Foster et al., 2018; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021). Most of these works deal with a finite action spaces and have regret scaling with the number of actions, which is necessary without further structural

assumptions (Agarwal et al., 2012). An exception is the works of Foster et al. (2020a) and Xu and Zeevi (2020), both of which consider the same setting as this chapter. Both of the algorithms in these works require solving subproblems based on maximizing quadratic forms (which is NP-hard in general (Sahni, 1974)), and cannot directly take advantage of the linear optimization oracle we consider. Also related is the work of Zhang (2021), which proposes a posterior sampling-style algorithm for the setting we consider. This algorithm is not fully comparable computationally, as it requires sampling from specific posterior distribution; it is unclear whether this can be achieved in a provably efficient fashion.

Linear contextual bandits. The linear contextual bandit problem is a special case of our setting in which $g^\star(x) = \theta \in \mathbb{R}^d$ is constant (that is, the reward function only depends on the context through the feature map ϕ). The most well-studied families of algorithms for this setting are UCB-style algorithms and posterior sampling. With a well-chosen prior and posterior distribution, posterior sampling can be implemented efficiently (Agrawal and Goyal, 2013), but it is unclear how to efficiently adapt this approach to accomodate general function approximation. Existing UCB-type algorithms require solving sub-problems based on maximizing quadratic forms, which is NP-hard in general (Sahni, 1974). One line of research aims to make UCB efficient by using hashing-based methods (MIPS) to approximate the maximum inner product (Yang et al., 2021; Jun et al., 2017). These methods have runtime sublinear (but still polynomial) in the number of actions.

Non-contextual linear bandits. For the problem of *non-contextual* linear bandits (with either stochastic or adversarial rewards), there is a long line of research on efficient algorithms that can take advantage of linear optimization oracles (Awerbuch and Kleinberg, 2008; McMahan and Blum, 2004; Dani and Hayes, 2006; Dani et al., 2008; Bubeck et al., 2012; Hazan and Karnin, 2016; Ito et al., 2019); see also work on the closely related problem of combinatorial pure exploration (Chen et al., 2017; Cao and Krishnamurthy, 2019; Katz-Samuels et al., 2020; Wagenmaker et al., 2021). In general, it is not clear how to lift these techniques to contextual bandits with

linearly-structured actions and general function approximation. We also mention that optimal design has been applied in the context of linear bandits, but these algorithms are restricted to the non-contextual setting (Lattimore and Szepesvári, 2020; Lattimore et al., 2020), or to pure exploration (Soare et al., 2014; Fiez et al., 2019). The only exception we are aware of is Ruan et al. (2021), who extend these developments to linear contextual bandits (i.e., where $g^\star(x) = \theta$), but critically use that contexts are stochastic.

Other approaches. Another line of research provides efficient contextual bandit methods under specific modeling assumptions on the context space or action space that differ from the ones we consider here. Zhou et al. (2020); Xu et al. (2020); Zhang et al. (2021); Kassraie and Krause (2022) provide generalizations of the UCB algorithm and posterior sampling based on the Neural Tangent Kernel (NTK). These algorithms can be used to learn context embeddings (i.e., $g(x)$) with general function approximation, but only lead to theoretical guarantees under strong RKHS-based assumptions. For large action spaces, these algorithms typically require enumeration over actions. Majzoubi et al. (2020) consider a setting with nonparametric action spaces and design an efficient tree-based learner; their guarantees, however, scale exponentially in the dimensionality of action space. Sen et al. (2021) provide heuristically-motivated but empirically-effective tree-based algorithms for contextual bandits with large action spaces, with theoretical guarantees when the actions satisfy certain tree-structured properties. Lastly, another empirically-successful approach is the policy gradient method (e.g., Williams (1992); Bhatnagar et al. (2009); Pan et al. (2019)). On the theoretical side, policy gradient methods do not address the issue of systematic exploration, and—to our knowledge—do not lead to provable guarantees for the setting considered in this chapter.

4.3 Warm-Up: Efficient Algorithms via Uniform Exploration

In this section, we present our first result: an efficient algorithm based on uniform exploration over a representative basis (SpannerGreedy; Algorithm 9). This algorithm achieves computational efficiency by taking advantage of an online regression oracle, but its regret bound has sub-optimal dependence on T. Beyond being practically useful in its own right, this result serves as a warm-up for Section 4.4.

Our algorithm is based on exploration with a *G-optimal design* for the embedding ϕ, which is a distribution over actions that minimizes a certain notion of worse-case variance (Kiefer and Wolfowitz, 1960; Atwood, 1969).

Definition 4.5 (G-optimal design)**.** *Let a set $\mathcal{Z} \subseteq \mathbb{R}^d$ be given. A distribution $q \in \Delta(\mathcal{Z})$ is said to be a G-optimal design with approximation factor $C_{\mathrm{opt}} \geqslant 1$ if*

$$\sup_{z \in \mathcal{Z}} \|z\|^2_{V(q)^{-1}} \leqslant C_{\mathrm{opt}} \cdot d,$$

where $V(q) := \mathbb{E}_{z \sim q}\left[zz^\top\right]$.

The following classical result guarantees existence of a G-optimal design.

Lemma 4.6 (Kiefer and Wolfowitz (1960))**.** *For any compact set $\mathcal{Z} \subseteq \mathbb{R}^d$, there exists an optimal design with $C_{\mathrm{opt}} = 1$.*

Algorithm 9 SpannerGreedy

Input: Exploration parameter $\varepsilon \in (0,1]$, online regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$, action optimization oracle $\mathbf{Alg}_{\mathsf{Opt}}$.

for $t = 1, 2, \ldots, T$ **do**
  Observe context x_t.
  Receive $\widehat{f}_t = f_{\widehat{g}_t}$ from regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
  Get $\widehat{a}_t \leftarrow \arg\max_{a \in \mathcal{A}} \langle \phi(x_t, a), \widehat{g}_t(x_t) \rangle$.
  Call subroutine to compute C_{opt}-approximate optimal design $q_t^{\mathrm{opt}} \in \Delta(\mathcal{A})$ for set $\{\phi(x_t, a)\}_{a \in \mathcal{A}}$.
    `// See Algorithm 13 for efficient solver.`
  Define $p_t := \varepsilon \cdot q_t^{\mathrm{opt}} + (1-\varepsilon) \cdot \mathbb{I}_{\widehat{a}_t}$.
  Sample $a_t \sim p_t$ and observe reward $r_t(a_t)$.
  Update oracle $\mathbf{Alg}_{\mathsf{Sq}}$ with $(x_t, a_t, r_t(a_t))$.

Algorithm 9 uses optimal design as a basis for exploration: At each round, the learner obtains an estimator $\widehat{f}_t$ from the regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$, then appeals to a subroutine to compute an (approximate) G-optimal design $q_t^{\mathrm{opt}} \in \Delta(\mathcal{A})$ for the action embedding $\{\phi(x_t, a)\}_{a \in \mathcal{A}}$. Fix an exploration parameter $\varepsilon > 0$, the algorithm then samples an action $a \sim q_t^{\mathrm{opt}}$ from the optimal design with probability ε ("exploration"), or plays the greedy action $\widehat{a}_t := \arg\max_{a \in \mathcal{A}} \widehat{f}_t(x_t, a)$ with probability $1-\varepsilon$ ("exploitation"). Algorithm 9 is efficient whenever an approximate optimal design can be computed efficiently, which can be achieved using Algorithm 13. We defer a detailed discussion of efficiency for a moment, and first state the main regret bound for the algorithm.

Theorem 4.7. *With a C_{opt}-approximate optimal design subroutine and an appropriate choice for $\varepsilon \in (0,1]$, Algorithm 9, with probability at least $1-\delta$, enjoys regret*

$$\mathbf{Reg}_{\mathsf{CB}}(T) = O\Big((C_{\mathrm{opt}} \cdot d)^{1/3} T^{2/3} (\mathbf{Reg}_{\mathsf{Sq}}(T) + \log(\delta^{-1}))^{1/3}\Big).$$

In particular, when invoked with Algorithm 13 (with $C = 2$) as a subroutine, the algorithm enjoys regret

$$\mathbf{Reg}_{\mathsf{CB}}(T) = O\Big(d^{2/3} T^{2/3} (\mathbf{Reg}_{\mathsf{Sq}}(T) + \log(\delta^{-1}))^{1/3}\Big).$$

and has per-round runtime $O(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Opt}} \cdot d^2 \log d + d^4 \log d)$ *and maximum memory* $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Opt}} + d^2)$.

Computational efficiency. The computational efficiency of Algorithm 9 hinges on the ability to efficiently compute an approximate optimal design (or, by convex duality, the John ellipsoid (John, 1948)) for the set $\{\phi(x_t, a)\}_{a \in \mathcal{A}}$. All off-the-shelf optimal design solvers that we are aware of require solving quadratic maximization subproblems, which in general cannot be reduced to a linear optimization oracle (Definition 4.3). While there are some special cases where efficient solvers exist (e.g., when $\mathcal{A}$ is a polytope (Cohen et al. (2019) and references therein)), computing an exact optimal design is NP-hard in general (Grötschel et al., 2012; Summa et al., 2014). To overcome this issue, we use the notion of a *barycentric spanner*, which acts as an approximate optimal design and can be computed efficiently using an action optimization oracle.

Definition 4.8 (Awerbuch and Kleinberg (2008))**.** *Let a compact set* $\mathcal{Z} \subseteq \mathbb{R}^d$ *of full dimension be given. For* $C \geqslant 1$*, a subset of points* $\mathcal{S} = \{z_1, \dots, z_d\} \subseteq \mathcal{Z}$ *is said to be a* C*-approximate barycentric spanner for* $\mathcal{Z}$ *if every point* $z \in \mathcal{Z}$ *can be expressed as a weighted combination of points in* $\mathcal{S}$ *with coefficients in* $[-C, C]$*.*

The following result shows that any barycentric spanner yields an approximate optimal design.

Lemma 4.9. *If* $\mathcal{S} = \{z_1, \dots, z_d\}$ *is a* C*-approximate barycentric spanner for* $\mathcal{Z} \subseteq \mathbb{R}^d$*, then* $q := \mathrm{unif}(\mathcal{S})$ *is a* $(C^2 \cdot d)$*-approximate optimal design.*

Using an algorithm introduced by Awerbuch and Kleinberg (2008), one can efficiently compute the C-approximate barycentric spanner for the set $\{\phi(x, a)\}_{a \in \mathcal{A}}$ using $O(d^2 \log_C d)$ calls to the action optimization oracle; their method is restated as Algorithm 13 in Section 4.7.1.

Key features of Algorithm 9. While the regret bound for Algorithm 9 scales with $T^{2/3}$, which is not optimal, this result constitutes the first computationally effi-

cient algorithm for contextual bandits with linearly structured actions and general function approximation. Additional features include:

- *Simplicity and practicality.* Appealing to uniform exploration makes Algorithm 9 easy to implement and highly practical. In particular, in the case where the action embedding does not depend on the context (i.e., $\phi(x, a) = \phi(a)$) an approximate design can be precomputed and reused, reducing the per-round runtime to $\widetilde{O}(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Opt}})$ and the maximum memory to $O(\mathcal{M}_{\mathsf{Sq}} + d)$.
- *Lifting optimal design to contextual bandits.* Previous bandit algorithms based on optimal design are limited to the non-contextual setting, and to pure exploration. Our result highlights for the first time that optimal design can be efficiently combined with general function approximation.

Proof sketch for Theorem 4.7. To analyze Algorithm 9, we follow a recipe introduced by Foster and Rakhlin (2020); Foster et al. (2021b) based on the *Decision-Estimation Coefficient* (DEC),[1] defined as $\mathsf{dec}_\gamma(\mathcal{F}) := \sup_{\widehat{f} \in \mathrm{conv}(\mathcal{F}), x \in \mathcal{X}} \mathsf{dec}_\gamma(\mathcal{F}; \widehat{f}, x)$, where

$$\mathsf{dec}_\gamma(\mathcal{F}; \widehat{f}, x) := \inf_{p \in \Delta(\mathcal{A})} \sup_{a^\star \in \mathcal{A}} \sup_{f^\star \in \mathcal{F}} \mathbb{E}_{a \sim p}\left[f^\star(x, a^\star) - f^\star(x, a) - \gamma \cdot (\widehat{f}(x, a) - f^\star(x, a))^2 \right]. \tag{4.3}$$

Foster et al. (2021b) consider a meta-algorithm which, at each round t, (i) computes $\widehat{f}_t$ by appealing to a regression oracle, (ii) computes a distribution $p_t \in \Delta(\mathcal{A})$ that solves the minimax problem in Eq. (4.3) with x_t and $\widehat{f}_t$ plugged in, and (iii) chooses the action a_t by sampling from this distribution. One can show (Lemma 4.16 in Section 4.7.1) that for any $\gamma > 0$, this strategy enjoys the following regret bound:

$$\mathbf{Reg}_{\mathsf{CB}}(T) \lesssim T \cdot \mathsf{dec}_\gamma(\mathcal{F}) + \gamma \cdot \mathbf{Reg}_{\mathsf{Sq}}(T), \tag{4.4}$$

[1] The original definition of the Decision-Estimation Coefficient in Foster et al. (2021b) uses Hellinger distance rather than squared error. The squared error version we consider here leads to tighter guarantees for bandit problems where the mean rewards serve as a sufficient statistic.

More generally, if one computes a distribution that does not solve Eq. (4.3) exactly, but instead certifies an upper bound on the DEC of the form $\mathsf{dec}_\gamma(\mathcal{F}) \leqslant \overline{\mathsf{dec}}_\gamma(\mathcal{F})$, the same result holds with $\mathsf{dec}_\gamma(\mathcal{F})$ replaced by $\overline{\mathsf{dec}}_\gamma(\mathcal{F})$. Algorithm 9 is a special case of this meta-algorithm, so to bound the regret it suffices to show that the exploration strategy in the algorithm certifies a bound on the DEC.

Lemma 4.10. *For any $\gamma \geqslant 1$, by choosing $\varepsilon = \sqrt{C_{\mathrm{opt}} \cdot d/4\gamma} \wedge 1$, the exploration strategy in Algorithm 9 certifies that $\mathsf{dec}_\gamma(\mathcal{F}) = O(\sqrt{C_{\mathrm{opt}} \cdot d/\gamma})$.*

Using Lemma 4.10, one can upper bound the first term in Eq. (4.4) by $O(T\sqrt{C_{\mathrm{opt}} d/\gamma})$. The regret bound in Theorem 4.7 follows by choosing γ to balance the two terms.

4.4 Efficient, Near-Optimal Algorithms

In this section we present SpannerIGW (Algorithm 10), an efficient algorithm with $\widetilde{O}(\sqrt{T})$ regret (Algorithm 10). We provide the algorithm and statistical guarantees in Section 4.4.1, then discuss computational efficiency in Section 4.4.2.

4.4.1 Algorithm and Statistical Guarantees

Building on the approach in Section 4.3, SpannerIGW uses the idea of exploration with an optimal design. However, in order to achieve $\sqrt{T}$ regret, we combine optimal design with the *inverse gap weighting* (IGW) technique. previously used in the finite-action contextual bandit setting (Abe and Long, 1999; Foster and Rakhlin, 2020).

Recall that for finite-action contextual bandits, the inverse gap weighting technique works as follows. Given a context x_t and estimator $\widehat{f}_t$ from the regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$, we assign a distribution to actions in $\mathcal{A}$ via the rule

$$p_t(a) := \frac{1}{\lambda + \gamma \cdot \left(\widehat{f}_t(x_t, \widehat{a}_t) - \widehat{f}_t(x_t, a)\right)},$$

where $\widehat{a}_t := \arg\max_{a \in \mathcal{A}} \widehat{f}_t(x_t, a)$ and $\lambda > 0$ is chosen such that $\sum_a p_t(a) = 1$. This strategy certifies that $\mathsf{dec}_\gamma(\mathcal{F}; \widehat{f}_t, x_t) \leqslant \frac{|\mathcal{A}|}{\gamma}$, which leads to regret $O\big(\sqrt{|\mathcal{A}| T \cdot \mathbf{Reg}_{\mathsf{Sq}}(T)}\big)$. While this is essentially optimal for the finite-action setting, the linear dependence on $|\mathcal{A}|$ makes it unsuitable for the large-action setting we consider.

To lift the IGW strategy to the large-action setting, Algorithm 10 combines it with optimal design with respect to a *reweighted embedding*. Let $\widehat{f} \in \mathcal{F}$ be given. For each action $a \in \mathcal{A}$, we define a reweighted embedding via

$$\overline{\phi}(x, a) := \frac{\phi(x, a)}{\sqrt{1 + \eta\Big(\widehat{f}(x, \widehat{a}) - \widehat{f}(x, a)\Big)}}, \tag{4.5}$$

where $\widehat{a} := \arg\max_{a \in \mathcal{A}} \widehat{f}(x, a)$ and $\eta > 0$ is a reweighting parameter to be tuned later. This reweighting is *action-dependent* since $\widehat{f}(x, a)$ term appears on the denominator. Within Algorithm 10, we compute a new reweighted embedding at each round $t \in [T]$ using $\widehat{f}_t = f_{\widehat{g}_t}$, the output of the regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.

Algorithm 10 proceeds by computing an optimal design $q_t^{\mathrm{opt}} \in \Delta(\mathcal{A})$ with respect to the reweighted embedding defined in Eq. (4.5). The algorithm then creates a distribution $q_t := \frac{1}{2} q_t^{\mathrm{opt}} + \frac{1}{2} \mathbb{I}_{\widehat{a}_t}$ by mixing the optimal design with a delta mass at the greedy action $\widehat{a}_t$. Finally, in Eq. (4.6), the algorithm computes an augmented version of the inverse gap weighting distribution by reweighting according to q_t. This approach certifies the following bound on the Decision-Estimation Coefficient.

Lemma 4.11. *For any $\gamma > 0$, by setting $\eta = \gamma/(C_{\mathrm{opt}} \cdot d)$, the exploration strategy used in Algorithm 10 certifies that $\mathsf{dec}_\gamma(\mathcal{F}) = O(C_{\mathrm{opt}} \cdot d/\gamma)$.*

This lemma shows that the reweighted IGW strategy enjoys the best of both worlds: By leveraging optimal design, we ensure good coverage for all actions, leading to $O(d)$ (rather than $O(|\mathcal{A}|)$) scaling, and by leveraging inverse gap weighting, we avoid excessive exploration, leading $O(1/\gamma)$ rather than $O(1/\sqrt{\gamma})$ scaling. Combining this result with Lemma 4.16 leads to our main regret bound for SpannerIGW.

Algorithm 10 SpannerIGW

Input: Exploration parameter $\gamma > 0$, online regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$, action optimization oracle $\mathbf{Alg}_{\mathsf{Opt}}$.

Define $\eta := \frac{\gamma}{C_{\mathrm{opt}} \cdot d}$.
for $t = 1, 2, \ldots, T$ **do**
  Observe context x_t.
  Receive $\widehat{f}_t = f_{\widehat{g}_t}$ from regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
  Get $\widehat{a}_t \leftarrow \arg\max_{a \in \mathcal{A}} \langle \phi(x_t, a), \widehat{g}_t(x_t) \rangle$.
  Call subroutine to compute C_{opt}-approximate optimal design $q_t^{\mathrm{opt}} \in \Delta(\mathcal{A})$ for reweighted embedding $\{\overline{\phi}(x_t, a)\}_{a \in \mathcal{A}}$ (Eq. (4.5) with $\widehat{f} = \widehat{f}_t$). // See Algorithm 11 for efficient solver.
  Define $q_t := \frac{1}{2} q_t^{\mathrm{opt}} + \frac{1}{2} \mathbb{I}_{\widehat{a}_t}$.
  For each $a \in \mathrm{supp}(q_t)$, define

$$p_t(a) := \frac{q_t(a)}{\lambda + \eta\left(\widehat{f}_t(x_t, \widehat{a}_t) - \widehat{f}_t(x_t, a)\right)}, \tag{4.6}$$

  where $\lambda \in [\frac{1}{2}, 1]$ is chosen so that $\sum_{a \in \mathrm{supp}(q_t)} p_t(a) = 1$.
  Sample $a_t \sim p_t$ and observe reward $r_t(a_t)$.
  Update $\mathbf{Alg}_{\mathsf{Sq}}$ with $(x_t, a_t, r_t(a_t))$.

Theorem 4.12. *Let $\delta \in (0, 1)$ be given. With a C_{opt}-approximate optimal design subroutine and an appropriate choice for $\gamma > 0$, Algorithm 10 ensures that with probability at least $1 - \delta$,*

$$\mathbf{Reg}_{\mathsf{CB}}(T) = O\Big(\sqrt{C_{\mathrm{opt}} \cdot d\, T \left(\mathbf{Reg}_{\mathsf{Sq}}(T) + \log(\delta^{-1})\right)} \Big).$$

In particular, when invoked with Algorithm 11 (with $C = 2$) as a subroutine, the algorithm has

$$\mathbf{Reg}_{\mathsf{CB}}(T) = O\Big(d \sqrt{T \left(\mathbf{Reg}_{\mathsf{Sq}}(T) + \log(\delta^{-1})\right)} \Big),$$

and has per-round runtime $O(\mathcal{T}_{\mathsf{Sq}} + (\mathcal{T}_{\mathsf{Opt}} \cdot d^3 + d^4) \cdot \log^2(\frac{T}{r}))$ and the maximum memory $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Opt}} + d^2 + d \log(\frac{T}{r}))$.

Algorithm 10 is the first computationally efficient algorithm with $\sqrt{T}$-regret for contextual bandits with general function approximation and linearly structured action spaces. In what follows, we show how to leverage the action optimization oracle (Definition 4.3) to achieve this efficiency.

4.4.2 Computational Efficiency

The computational efficiency of Algorithm 10 hinges on the ability to efficiently compute an optimal design. As with Algorithm 9, we address this issue by appealing to the notion of a barycentric spanner, which serves as an approximate optimal design. However, compared to Algorithm 9, a substantial additional challenge is that Algorithm 10 requires an approximate optimal design for the *reweighted* embeddings. Since the reweighting is action-dependent, the action optimization oracle $\mathbf{Alg}_{\mathsf{Opt}}$ cannot be directly applied to optimize over the reweighted embeddings, which prevents us from appealing to an out-of-the-box solver (Algorithm 13) in the same fashion as the prequel.

Algorithm 11 ReweightedSpanner

Input: Context $x \in \mathcal{X}$, oracle prediction $\widehat{g}(x) \in \mathbb{R}^d$, action $\widehat{a} := \arg\max_{a \in \mathcal{A}} \langle \phi(x,a), \widehat{g}(x) \rangle$, reweighting parameter $\eta > 0$, approximation factor $C > \sqrt{2}$, initial set $\mathcal{S} = (a_1, \dots, a_d)$ with $|\det(\phi(x,\mathcal{S}))| \geqslant r^d$ for $r \in (0,1)$.

1: **while** not break **do**
2: **for** $i = 1, \dots, d$ **do**
3: Compute $\theta \in \mathbb{R}^d$ representing linear function $\overline{\phi}(x,a) \mapsto \det(\overline{\phi}(x, \mathcal{S}_i(a)))$, where $\mathcal{S}_i(a) := (a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_d)$. `// $\overline{\phi}$ is computed from $f_{\widehat{g}}$, $\widehat{a}$, and $\eta$ via Eq. (4.5).`
4: Get $a \leftarrow$ IGW-ArgMax$(\theta; x, \widehat{g}(x), \eta, r)$. `// Algorithm 12.`
5: **if** $\left|\det(\overline{\phi}(x, \mathcal{S}_i(a)))\right| \geqslant \frac{\sqrt{2}C}{2} \left|\det(\overline{\phi}(x, \mathcal{S}))\right|$ **then**
6: Update $a_i \leftarrow a$.
7: **continue** to line 2.
8: **break**
9: **return** C-approximate barycentric spanner $\mathcal{S}$.

To address the challenges above, we introduce ReweightedSpanner (Algorithm 11),

a barycentric spanner computation algorithm which is tailored to the reweighted embedding $\overline{\phi}$. To describe the algorithm, let us introduce some additional notation. For a set $\mathcal{S} \subseteq \mathcal{A}$ of d actions, we let $\det(\overline{\phi}(x, \mathcal{S}))$ denote the determinant of the d-by-d matrix whose columns are $\{\overline{\phi}(x, a)\}_{a \in \mathcal{A}}$. ReweightedSpanner adapts the barycentric spanner computation approach of Awerbuch and Kleinberg (2008), which aims to identify a subset $\mathcal{S} \subseteq \mathcal{A}$ with $|\mathcal{S}| = d$ that approximately maximizes $|\det(\overline{\phi}(x, \mathcal{S}))|$. The key feature of ReweightedSpanner is a subroutine, IGW-ArgMax (Algorithm 12), which implements an (approximate) action optimization oracle for the reweighted embedding:

$$\arg\max_{a \in \mathcal{A}} \langle \overline{\phi}(x, a), \theta \rangle. \tag{4.7}$$

IGW-ArgMax uses line search reduce the problem in Eq. (4.7) to a sequence of linear optimization problems with respect to the *unweighted* embeddings, each of which can be solved using $\mathbf{Alg}_{\mathsf{Opt}}$. This yields the following guarantee for Algorithm 11.

Theorem 4.13. *Suppose that Algorithm 11 is invoked with parameters $\eta > 0$, $r \in (0, 1)$, and $C > \sqrt{2}$, and that the initialization set $\mathcal{S}$ satisfies $|\det(\phi(x, \mathcal{S}))| \geqslant r^d$. Then the algorithm returns a C-approximate barycentric spanner with respect to the reweighted embedding set $\{\overline{\phi}(x, a)\}_{a \in \mathcal{A}}$, and does so with $O((\mathcal{T}_{\mathsf{Opt}} \cdot d^3 + d^4) \cdot \log^2(e \vee \frac{\eta}{r}))$ runtime and $O(\mathcal{M}_{\mathsf{Opt}} + d^2 + d \log(e \vee \frac{\eta}{r}))$ memory.*

We refer to Section 4.7.3.1 for self-contained analysis of IGW-ArgMax.

Algorithm 12 IGW-ArgMax

Input: Linear parameter $\theta \in \mathbb{R}^d$, context $x \in \mathcal{X}$, oracle prediction $\widehat{g}(x) \in \mathbb{R}^d$, reweighting parameter $\eta > 0$, initialization constant $r \in (0, 1)$.

Define $N := \lceil d \log_{\frac{4}{3}}(\frac{2\eta+1}{r}) \rceil$.
Define $\mathcal{E} := \{(\frac{3}{4})^i\}_{i=1}^N \cup \{-(\frac{3}{4})^i\}_{i=1}^N$.
Initialize $\widehat{\mathcal{A}} = \emptyset$.
for each $\varepsilon \in \mathcal{E}$ **do**
  Compute $\bar{\theta} \leftarrow 2\varepsilon\theta + \varepsilon^2\eta \cdot \widehat{g}(x)$.
  Get $a \leftarrow \arg\max_{a \in \mathcal{A}} \langle \phi(x, a), \bar{\theta} \rangle$; add a to $\widehat{\mathcal{A}}$.
return $\arg\max_{a \in \widehat{\mathcal{A}}} \langle \overline{\phi}(x, a), \theta \rangle^2$ // $\widetilde{O}(d)$ candidates.

On the initialization requirement. The runtime for Algorithm 11 scales with $\log(r^{-1})$, where $r \in (0,1)$ is such that $\det(\phi(x,\mathcal{S})) \geqslant r^d$ for the initial set $\mathcal{S}$. In Section 4.7.3.3, we provide computationally efficient algorithms for initialization under various assumptions on the action space.

4.5 Empirical Results

In this section we investigate the empirical performance of SpannerGreedy and SpannerIGW through three experiments. First, we compare the spanner-based algorithms to state-of-the art finite-action algorithms on a large-action dataset; this experiment features nonlinear, learned context embeddings $g \in \mathcal{G}$. Next, we study the impact of redundant actions on the statistical performance of said algorithms. Finally, we experiment with a large-scale large-action contextual bandit benchmark, where we find that the spanner-based methods exhibit excellent performance.

Preliminaries. We conduct experiments on three datasets, whose details are summarized in Table 4.1. oneshotwiki (Singh et al., 2012; Vasnetsov, 2018) is a named-entity recognition task where contexts are text phrases preceding and following the mention text, and where actions are text phrases corresponding to the concept names. amazon-3m (Bhatia et al., 2016) is an extreme multi-label dataset whose contexts are text phrases corresponding to the title and description of an item, and whose actions are integers corresponding to item tags. Actions are embedded into $\mathbb{R}^d$ with d specified in Table 4.1. We construct binary rewards for each dataset, and report 90% bootstrap confidence intervals (CIs) of the rewards in the experiments. We defer other experimental details to Section 4.7.4.1. Code to reproduce all results is available at `https://github.com/pmineiro/linrepcb`.

Comparison with finite-action baselines. We compare SpannerGreedy and SpannerIGW with their finite-action counterparts ε-Greedy and SquareCB (Foster and Rakhlin, 2020) on the oneshotwiki-14031 dataset. We consider *bilinear models* in which regression functions take the form $f(x,a) = \langle \phi(a), Wx \rangle$ where W is a matrix

Table 4.1: Details of datasets used in experiments.

Dataset	T	$\lvert\mathcal{A}\rvert$	d
oneshotwiki-311	622000	311	50
oneshotwiki-14031	2806200	14031	50
amazon-3m	1717899	2812281	800

of learned parameters; the *deep models* of the form $f(x, a) = \langle \phi(a), W\bar{g}(x) \rangle$, where $\bar{g}$ is a learned two-layer neural network and W contains learned parameters as before.[2] Table 4.2 presents our results. We find that SpannerIGW performs best, and that both spanner-based algorithms either tie or exceed their finite-action counterparts. In addition, we find that working with deep models uniformly improves performance for all methods. We refer to Table 4.4 in Section 4.7.4.3 for timing information.

Table 4.2: Comparison on oneshotwiki-14031. Values are the average progressive rewards (confidence intervals), scaled by 1000. We include the performance of the best constant predictor (as a baseline) and the supervised learner (as a skyline).

Algorithm	Regression Function	
	Bilinear	Deep
best constant	0.07127	
ε-Greedy	[5.00, 6.27]	[7.15, 8.52]
SpannerGreedy	[6.29, 7.08]	[6.67, 8.30]
SquareCB	[7.57, 8.59]	[10.4, 11.3]
SpannerIGW	[8.84, 9.68]	[11.2, 12.2]
supervised	[31.2, 31.3]	[36.7, 36.8]

Impact of redundancy. Finite-action contextual bandit algorithms can explore excessively in the presence of redundant actions. To evaluate performance in the face of redundancy, we augment oneshotwiki-311 by duplicating action the final action. Table 4.3 displays the performance of SpannerIGW and its finite-action counterpart, SquareCB, with a varying number of duplicates. We find that SpannerIGW

[2] Also see Section 4.7.4.1 for details.

is completely invariant to duplicates (in fact, the algorithm produces numerically identical output when the random seed is fixed), but SquareCB is negatively impacted and over-explores the duplicated action. SpannerGreedy and ε-Greedy behave analogously (not shown).

Table 4.3: Redundancy study on oneshotwiki-311. Values are the average progressive rewards (confidence intervals), scaled by 100.

Duplicates	SpannerIGW	SquareCB
0	[12.6, 13.0]	[12.2, 12.6]
16	[12.6, 13.0]	[12.1, 12.4]
256	[12.6, 13.0]	[10.2, 10.6]
1024	[12.6, 13.0]	[8.3, 8.6]

Large scale exhibition. We conduct a large scale experiment using the amazon-3m dataset. Following Sen et al. (2021), we study the top-k setting where k actions are selected at each round. Out of the total number of actions sampled, we let r denote the number of actions sampled for exploration. We apply SpannerGreedy for this dataset and consider regression functions similar to the deep models discussed before. The setting $(k = 1)$ corresponds to running our algorithm unmodified, and $(k = 5, r = 3)$ corresponds to selecting 5 actions per round and using 3 exploration slots. Fig. 4.1 in Section 4.7.4.4 displays the results. For $(k = 1)$ the final CI is $[0.1041, 0.1046]$, and for $(k = 5, r = 3)$ the final CI is $[0.438, 0.440]$.

In the setup with $(k = 5, r = 3)$, our results are directly comparable to Sen et al. (2021), who evaluated a tree-based contextual bandit method on the same dataset. The best result from Sen et al. (2021) achieves roughly 0.19 reward with $(k = 5, r = 3)$, which we exceed by a factor of 2. This indicates that our use of embeddings provides favorable inductive bias for this problem, and underscores the broad utility of our techniques (which leverage embeddings). For $(k = 5, r = 3)$, our inference time on a commodity CPU with batch size 1 is 160ms per example, which is slower than the time of 7.85ms per example reported in Sen et al. (2021).

4.6 Discussion

We provide the first efficient algorithms for contextual bandits with continuous, linearly structured action spaces and general-purpose function approximation. We highlight some natural directions for future research below.

- **Efficient algorithms for nonlinear action spaces.** Our algorithms take advantage of linearly structured action spaces by appealing to optimal design. Can we develop computationally efficient methods for contextual bandits with nonlinear dependence on the action space?
- **Reinforcement learning.** The contextual bandit problem is a special case of the reinforcement learning problem with horizon one. Given our positive results in the contextual bandit setting, a natural next step is to extend our methods to reinforcement learning problems with large action/decision spaces. For example, Foster et al. (2021b) build on our computational tools to provide efficient algorithms for reinforcement learning with bilinear classes.

Beyond these directions, natural domains in which to extend our techniques include pure exploration and off-policy learning with linearly structured actions.

4.7 Proofs and Supporting Results

4.7.1 Proofs and Supporting Results for Section 4.3

This section is organized as follows. We provide supporting results in Section 4.7.1.1, then give the proof of Theorem 4.7 in Section 4.7.1.2.

4.7.1.1 Supporting Results

Barycentric Spanner and Optimal Design.

Algorithm 13 restates an algorithm of Awerbuch and Kleinberg (2008), which efficiently computes a barycentric spanner (Definition 4.8) given access to a linear optimization oracle (Definition 4.3). Recall that, for a set $\mathcal{S} \subset \mathcal{A}$ of d actions, the notation $\det(\overline{\phi}(x, \mathcal{S}))$ (resp. $\det(\phi(x, \mathcal{S}))$) denotes the determinant of the d-by-d matrix whose columns are the $\overline{\phi}$ (resp. ϕ) embeddings of actions.

Algorithm 13 Approximate Barycentric Spanner (Awerbuch and Kleinberg, 2008)

Input: Context $x \in \mathcal{X}$ and approximation factor $C > 1$.

1: **for** $i = 1, \dots, d$ **do**
2: Compute $\theta \in \mathbb{R}^d$ representing linear function $\phi(x, a) \mapsto \det(\phi(x, a_1), \dots, \phi(x, a_{i-1}), \phi(x, a), e_{i+1}, \dots, e_d)$.
3: Get $a_i \leftarrow \arg\max_{a \in \mathcal{A}} |\langle \phi(x, a), \theta \rangle|$.
4: Construct $\mathcal{S} = (a_1, \dots, a_d)$. `// Initial set of actions` $\mathcal{S} \subseteq \mathcal{A}$ `such that` $|\mathcal{S}| = d$ `and` $|\det(\phi(x, \mathcal{S}))| > 0$.
5: **while** not break **do**
6: **for** $i = 1, \dots, d$ **do**
7: Compute $\theta \in \mathbb{R}^d$ representing linear function $\phi(x, a) \mapsto \det(\phi(x, \mathcal{S}_i(a)))$, where $\mathcal{S}_i(a) := (a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_d)$.
8: Get $a \leftarrow \arg\max_{a \in \mathcal{A}} |\langle \phi(x, a), \theta \rangle|$.
9: **if** $|\det(\phi(x, \mathcal{S}_i(a)))| \geqslant C|\det(\phi(x, \mathcal{S}))|$ **then**
10: Update $a_i \leftarrow a$.
11: **continue** to line 5.
12: **break**
13: **return** C-approximate barycentric spanner $\mathcal{S}$.

Lemma 4.14 (Awerbuch and Kleinberg (2008))**.** *For any $x \in \mathcal{X}$, Algorithm 13 computes a C-approximate barycentric spanner for $\{\phi(x, a) : a \in \mathcal{A}\}$ within $O(d \log_C d)$ iterations of the while-loop.*

Lemma 4.15. *Fix any constant $C > 1$. Algorithm 13 can be implemented with runtime $O(\mathcal{T}_{\mathsf{Opt}} \cdot d^2 \log d + d^4 \log d)$ and memory $O(\mathcal{M}_{\mathsf{Opt}} + d^2)$.*

Proof of Lemma 4.15. We provide the computational complexity analysis starting from the while-loop (line 5-12) in the following. The computational complexity regarding the first for-loop (line 1-3) can be similarly analyzed.

- *Outer loops (lines 5-6).* From Lemma 4.14, we know that Algorithm 13 terminates within $O(d \log d)$ iterations of the while-loop (line 5). It is also clear that the for-loop (line 6) is invoked at most d times.

- *Computational complexity for lines 7-10.* We discuss how to efficiently implement this part using rank-one updates. We analyze the computational complexity for each line in the following.

  - *Line 7.* We discuss how to efficiently compute the linear function θ through rank-one updates. Fix any $Y \in \mathbb{R}^d$. Let Φ_S denote the invertible (by construction) matrix whose k-th column is $\phi(x, a_k)$ (with $a_k \in S$). Using the rank-one update formula for the determinant (Meyer, 2000), we have

$$\begin{aligned}
&\det(\phi(x, a_1), \ldots, \phi(x, a_{i-1}), Y, \phi(x, a_{i+1}), \ldots, \phi(x, a_d)) \\
&= \det\Big(\Phi_S + \big(Y - \phi(x, a_i)\big) e_i^\top\Big) \\
&= \det(\Phi_S) \cdot \Big(1 + e_i^\top \Phi_S^{-1} \big(Y - \phi(x, a_i)\big)\Big) \\
&= \big\langle Y, \det(\Phi_S) \cdot \big(\Phi_S^{-1}\big)^\top e_i \big\rangle + \det(\Phi_S) \cdot \big(1 - e_i^\top \Phi_S^{-1} \phi(x, a_i)\big). \qquad (4.8)
\end{aligned}$$

    We first notice that $\det(\Phi_S) \cdot \big(1 - e_i^\top \Phi_S^{-1} \phi(x, a_i)\big) = 0$ since one can take $Y = 0 \in \mathbb{R}^d$. We can then write

$$\det(\phi(x, a_1), \ldots, \phi(x, a_{i-1}), Y, \phi(x, a_{i+1}), \ldots, \phi(x, a_d)) = \langle Y, \theta \rangle$$

    where $\theta = \det(\Phi_S) \cdot \big(\Phi_S^{-1}\big)^\top e_i$. Thus, whenever $\det(\Phi_S)$ and Φ_S^{-1} are known, compute θ takes $O(d)$ time. The maximum memory requirement is $O(d^2)$, following from the storage of Φ_S^{-1}.

  - *Line 8.* When θ is computed, we can compute a by first compute $a_+ := \arg\max_{a \in \mathcal{A}} \langle \phi(x, a), \theta \rangle$ and $a_- := \arg\max_{a \in \mathcal{A}} -\langle \phi(x, a), \theta \rangle$ and then compare the two. This process takes two oracle calls to $\mathbf{Alg}_{\mathsf{Opt}}$, which takes $O(\mathcal{T}_{\mathsf{Opt}})$ time. The maximum memory requirement is $O(\mathcal{M}_{\mathsf{Opt}} + d)$,

following from the memory requirement of $\mathbf{Alg}_{\mathsf{Opt}}$ and the storage of θ.

- *Line 9.* Once θ and $\det(\Phi_S)$ are computed, checking the updating criteria takes $O(d)$ time. The maximum memory requirement is $O(d)$, following from the storage of $\phi(x, a)$ and θ.
- *Line 10.* We discuss how to efficiently update $\det(\Phi_S)$ and Φ_S^{-1} through rank-one updates. If an update $a_i = a$ is made, we can update the determinant using rank-one update (as in Eq. (4.8)) with runtime $O(d)$ and memory $O(d^2)$; and update the inverse matrix using the Sherman-Morrison rank-one update formula (Sherman and Morrison, 1950), i.e.,

$$\left(\Phi_S + \big(\phi(x,a) - \phi(x,a_i)\big)e_i^\top\right)^{-1} = \Phi_S^{-1} - \frac{\Phi_S^{-1}\big(\phi(x,a) - \phi(x,a_i)\big)e_i^\top \Phi_S^{-1}}{1 + e_i \Phi_S^{-1}\big(\phi(x,a) - \phi(x,a_i)\big)},$$

  which can be implemented in $O(d^2)$ time and memory. Note that the updated matrix must be invertible by construction.

Thus, using rank-one updates, the total runtime adds up to $O(\mathcal{T}_{\mathsf{Opt}} + d^2)$ and the maximum memory requirement is $O(\mathcal{M}_{\mathsf{Opt}} + d^2)$. We also remark that the initial matrix determinant and inverse can be computed cheaply since the first iteration of the first for-loop (i.e., line 2 with $i = 1$) is updated from the identity matrix.

To summarize, Algorithm 13 has runtime $O(\mathcal{T}_{\mathsf{Opt}} \cdot d^2 \log d + d^4 \log d)$ and uses at most $O(\mathcal{M}_{\mathsf{Opt}} + d^2)$ units of memory. □

The next proposition shows that a barycentric spanner implies an approximate optimal design. The result is well-known (e.g., Hazan and Karnin (2016)), but we provide a proof here for completeness.

Lemma 4.9. *If $S = \{z_1, \ldots, z_d\}$ is a C-approximate barycentric spanner for $\mathcal{Z} \subseteq \mathbb{R}^d$, then $q := \mathrm{unif}(S)$ is a $(C^2 \cdot d)$-approximate optimal design.*

Proof of Lemma 4.9. Assume without loss of generality that $\mathcal{Z} \subseteq \mathbb{R}^d$ spans $\mathbb{R}^d$. By Definition 4.8, we know that for any $z \in \mathcal{Z}$, we can represent z as a weighted sum of

elements in $\mathcal{S}$ with coefficients in the range $[-C, C]$. Let $\Phi_{\mathcal{S}} \in \mathbb{R}^{d \times d}$ be the matrix whose columns are the vectors in $\mathcal{S}$. For any $z \in \mathcal{Z}$, we can find $\theta \in [-C, C]^d$ such that $z = \Phi_{\mathcal{S}}\theta$. Since $\Phi_{\mathcal{S}}$ is invertible (by construction), we can write $\theta = \Phi_{\mathcal{S}}^{-1}z$, which implies the result via

$$C^2 \cdot d \geqslant \|\theta\|_2^2 = \|z\|^2_{(\Phi_{\mathcal{S}}\Phi_{\mathcal{S}}^\top)^{-1}} = \frac{1}{d} \cdot \|z\|^2_{V(q)^{-1}}.$$

□

Regret Decomposition.

Fix any $\gamma > 0$. We consider the following meta algorithm that utilizes the online regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$ defined in Assumption 4.2.

For $t = 1, 2, \ldots, T$:

- Get context $x_t \in \mathcal{X}$ from the environment and regression function $\widehat{f}_t \in \mathrm{conv}(\mathcal{F})$ from the online regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
- Identify the distribution $p_t \in \Delta(\mathcal{A})$ that solves the minimax problem $\mathsf{dec}_\gamma(\mathcal{F}; \widehat{f}_t, x_t)$ (defined in Eq. (4.3)) and play action $a_t \sim p_t$.
- Observe reward r_t and update regression oracle with example (x_t, a_t, r_t).

The following result bounds the contextual bandit regret for the meta algorithm described above. The result is a variant of the regret decomposition based on the Decision-Estimation Coefficient given in Foster et al. (2021b), which generalizes Foster and Rakhlin (2020). The slight differences in constant terms are due to the difference in reward range.

Lemma 4.16 (Foster and Rakhlin (2020); Foster et al. (2021b))**.** *Suppose that Assumption 4.2 holds. Then probability at least $1 - \delta$, the contextual bandit regret is upper bounded as follows:*

$$\mathbf{Reg}_{\mathsf{CB}}(T) \leqslant \mathsf{dec}_\gamma(\mathcal{F}) \cdot T + 2\gamma \cdot \mathbf{Reg}_{\mathsf{Sq}}(T) + 64\gamma \cdot \log(2\delta^{-1}) + \sqrt{8T\log(2\delta^{-1})}.$$

In general, identifying a distribution that *exactly* solves the minimax problem corresponding to the DEC may be impractical. However, if one can identify a distribution that instead certifies an *upper bound* $\overline{\mathsf{dec}}_\gamma(\mathcal{F})$ on the Decision-Estimation Coefficient (in the sense that $\mathsf{dec}_\gamma(\mathcal{F}) \leqslant \overline{\mathsf{dec}}_\gamma(\mathcal{F})$), the regret bound in Lemma 4.16 continues to hold with $\mathsf{dec}_\gamma(\mathcal{F})$ replaced by $\overline{\mathsf{dec}}_\gamma(\mathcal{F})$.

Proof of Lemma 4.10.

Lemma 4.10. *For any $\gamma \geqslant 1$, by choosing $\varepsilon = \sqrt{C_{\mathrm{opt}} \cdot d/4\gamma} \wedge 1$, the exploration strategy in Algorithm 9 certifies that $\mathsf{dec}_\gamma(\mathcal{F}) = O(\sqrt{C_{\mathrm{opt}} \cdot d/\gamma})$.*

Proof of Lemma 4.10. Fix a context $x \in \mathcal{X}$. In our setting, where actions are linearly structured, we can equivalently write the Decision-Estimation Coefficient $\mathsf{dec}_\gamma(\mathcal{F}; \widehat{f}, x)$ as

$$\mathsf{dec}_\gamma(\mathcal{G}; \widehat{g}, x) := \inf_{p \in \Delta(\mathcal{A})} \sup_{a^\star \in \mathcal{A}} \sup_{g^\star \in \mathcal{G}} \mathbb{E}_{a\sim p}\left[\left\langle \phi(x, a^\star) - \phi(x, a), g^\star(x)\right\rangle - \gamma \cdot \left(\left\langle \phi(x, a), g^\star(x) - \widehat{g}(x)\right\rangle\right)^2\right]. \tag{4.9}$$

Recall that within our algorithms, $\widehat{g} \in \mathrm{conv}(\mathcal{G})$ is obtained from the estimator $\widehat{f} = f_{\widehat{g}}$ output by $\mathbf{Alg}_{\mathsf{Sq}}$. We will bound the quantity in Eq. (4.9) uniformly for all $x \in \mathcal{X}$ and $\widehat{g} : \mathcal{X} \to \mathbb{R}^d$ with $\|\widehat{g}\| \leqslant 1$. Recall that we assume $\sup_{g\in\mathcal{G}, x\in\mathcal{X}} \|g(x)\| \leqslant 1$.

Denote $\widehat{a} := \arg\max_{a\in\mathcal{A}} \langle \phi(x, a), \widehat{g}(x)\rangle$ and $a^\star := \arg\max_{a\in\mathcal{A}} \langle \phi(x, a), g^\star(x)\rangle$. For any $\varepsilon \leqslant 1$, let $p := \varepsilon \cdot q^{\mathrm{opt}} + (1-\varepsilon) \cdot \mathbb{I}_{\widehat{a}}$, where $q^{\mathrm{opt}} \in \Delta(\mathcal{A})$ is any C_{opt}-approximate optimal design for the embedding $\{\phi(x, a)\}_{a\in\mathcal{A}}$. We have the following decomposition.

$$\begin{aligned}\mathbb{E}_{a\sim p}\left[\left\langle \phi(x, a^\star) - \phi(x, a), g^\star(x)\right\rangle\right] &= \mathbb{E}_{a\sim p}\left[\left\langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x)\right\rangle\right] \\ &\quad + \mathbb{E}_{a\sim p}\left[\left\langle \phi(x, a), \widehat{g}(x) - g^\star(x)\right\rangle\right] \\ &\quad + \Big(\left\langle \phi(x, a^\star), g^\star(x)\right\rangle - \left\langle \phi(x, \widehat{a}), \widehat{g}(x)\right\rangle\Big).\end{aligned} \tag{4.10}$$

For the first term in Eq. (4.10), we have

$$\begin{aligned}\mathbb{E}_{a\sim p}\left[\left\langle \phi(x,\widehat{a})-\phi(x,a),\widehat{g}(x)\right\rangle\right] &= \varepsilon\cdot\mathbb{E}_{a\sim q^{\mathrm{opt}}}\left[\left\langle \phi(x,\widehat{a})-\phi(x,a),\widehat{g}(x)\right\rangle\right]\\ &\leqslant 2\varepsilon\cdot\sup_{x\in\mathcal{X},a\in\mathcal{A}}\|\phi(x,a)\|\cdot\sup_{x\in\mathcal{X}}\|\widehat{g}(x)\|\\ &\leqslant 2\varepsilon.\end{aligned}$$

Next, since

$$\left\langle \phi(x,a),\widehat{g}(x)-g^\star(x)\right\rangle \leqslant \frac{\gamma}{2}\cdot\left(\left\langle \phi(x,a),\widehat{g}(x)-g^\star(x)\right\rangle\right)^2+\frac{1}{2\gamma}$$

by AM-GM inequality, we can bound the second term in Eq. (4.10) by

$$\mathbb{E}_{a\sim p}\left[\left\langle \phi(x,a),\widehat{g}(x)-g^\star(x)\right\rangle\right] \leqslant \frac{\gamma}{2}\cdot\mathbb{E}_{a\sim p}\left[\left(\left\langle \phi(x,a),\widehat{g}(x)-g^\star(x)\right\rangle\right)^2\right]+\frac{1}{2\gamma}.$$

We now turn our attention to the third term. Observe that since $\widehat{a}$ is optimal for $\widehat{g}$, $\left\langle \phi(x,\widehat{a}),\widehat{g}(x)\right\rangle \geqslant \left\langle \phi(x,a^\star),\widehat{g}(x)\right\rangle$. As a result, defining $V(q^{\mathrm{opt}}) := \mathbb{E}_{a\sim q^{\mathrm{opt}}}[\phi(x,a)\phi(x,a)^\top]$, we have

$$\begin{aligned}\left\langle \phi(x,a^\star),g^\star(x)\right\rangle-\left\langle \phi(x,\widehat{a}),\widehat{g}(x)\right\rangle &\leqslant \left\langle \phi(x,a^\star),g^\star(x)-\widehat{g}(x)\right\rangle\\ &\leqslant \left\|\phi(x,a^\star)\right\|_{V(q^{\mathrm{opt}})^{-1}}\cdot\left\|g^\star(x)-\widehat{g}(x)\right\|_{V(q^{\mathrm{opt}})}\\ &= \frac{1}{2\gamma\varepsilon}\cdot\left\|\phi(x,a^\star)\right\|^2_{V(q^{\mathrm{opt}})^{-1}}\\ &\quad+\frac{\gamma}{2}\cdot\varepsilon\cdot\mathbb{E}_{a\sim q^{\mathrm{opt}}}\left[\left(\phi(x,a),g^\star(x)-\widehat{g}(x)\right)^2\right]\\ &\leqslant \frac{C_{\mathrm{opt}}\cdot d}{2\gamma\varepsilon}+\frac{\gamma}{2}\cdot\mathbb{E}_{a\sim p}\left[\left(\phi(x,a),g^\star(x)-\widehat{g}(x)\right)^2\right].\end{aligned}$$

Here, the third line follows from the AM-GM inequality, and the last line follows from the (C_{opt}-approximate) optimal design property and the definition of p.

Combining these bounds, we have

$$\mathsf{dec}_\gamma(\mathcal{F}) = \inf_{p\in\Delta(\mathcal{A})} \sup_{a^\star\in\mathcal{A}} \sup_{g^\star\in\mathcal{G}} \mathsf{dec}_\gamma(\mathcal{G}; \widehat{g}, x) \leqslant 2\varepsilon + \frac{1}{2\gamma} + \frac{C_{\mathrm{opt}}\cdot d}{2\gamma\varepsilon}.$$

Since $\gamma \geqslant 1$, taking $\varepsilon := \sqrt{C_{\mathrm{opt}}\cdot d/4\gamma} \wedge 1$ gives

$$\mathsf{dec}_\gamma(\mathcal{F}) \leqslant 2\sqrt{\frac{C_{\mathrm{opt}}\cdot d}{\gamma}} + \frac{1}{2\gamma} \leqslant 3\sqrt{\frac{C_{\mathrm{opt}}\cdot d}{\gamma}}$$

whenever $\varepsilon < 1$. On the other hand, when $\varepsilon = 1$, this bound holds trivially. □

4.7.1.2 Proof of Theorem 4.7

Theorem 4.7. *With a C_{opt}-approximate optimal design subroutine and an appropriate choice for $\varepsilon \in (0, 1]$, Algorithm 9, with probability at least $1-\delta$, enjoys regret*

$$\mathbf{Reg}_{\mathsf{CB}}(T) = O\Big((C_{\mathrm{opt}}\cdot d)^{1/3}T^{2/3}(\mathbf{Reg}_{\mathsf{Sq}}(T) + \log(\delta^{-1}))^{1/3}\Big).$$

In particular, when invoked with Algorithm 13 (with $C = 2$) as a subroutine, the algorithm enjoys regret

$$\mathbf{Reg}_{\mathsf{CB}}(T) = O\Big(d^{2/3}T^{2/3}(\mathbf{Reg}_{\mathsf{Sq}}(T) + \log(\delta^{-1}))^{1/3}\Big).$$

and has per-round runtime $O(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Opt}}\cdot d^2\log d + d^4\log d)$ and maximum memory $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Opt}} + d^2)$.

Proof of Theorem 4.7. Consider $\gamma \geqslant 1$. Combining Lemma 4.10 with Lemma 4.16, we have

$$\mathbf{Reg}_{\mathsf{CB}}(T) \leqslant 3T\cdot\sqrt{\frac{C_{\mathrm{opt}}\cdot d}{\gamma}} + 2\gamma\cdot\mathbf{Reg}_{\mathsf{Sq}}(T) + 64\gamma\cdot\log(2\delta^{-1}) + \sqrt{8T\log(2\delta^{-1})}.$$

The regret bound in Theorem 4.7 immediately follows by choosing

$$\gamma = \left(\frac{3T\sqrt{C_{\mathrm{opt}} \cdot d}}{2\mathbf{Reg}_{\mathsf{Sq}}(T) + 64\log(2\delta^{-1})} \right)^{2/3} \vee 1.$$

In particular, when Algorithm 13 is invoked as a subroutine with parameter $C = 2$, Lemma 4.9 implies that we may take $C_{\mathrm{opt}} \leqslant 4d$.

Computational complexity. We now bound the per-round computational complexity of Algorithm 9 when Algorithm 13 is used as a subroutine to compute the approximate optimal design. Outside of the call to Algorithm 13, Algorithm 9 uses $O(1)$ calls to $\mathbf{Alg}_{\mathsf{Sq}}$ to obtain $\widehat{g}_t(x_t) \in \mathbb{R}^d$ and to update $\widehat{f}_t$, and uses a single call to $\mathbf{Alg}_{\mathsf{Opt}}$ to compute $\widehat{a}_t$. With the optimal design q_t^{opt} returned by Algorithm 13 (represented as a barycentric spanner), sampling from p_t takes at most $O(d)$ time, since $|\mathrm{supp}(p_t)| \leqslant d+1$. outside of Algorithm 13 adds up to $O(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Opt}} + d)$. In terms of memory, calling $\mathbf{Alg}_{\mathsf{Sq}}$ and $\mathbf{Alg}_{\mathsf{Opt}}$ takes $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Opt}})$ units, and maintaining the distribution p_t (the barycentric spanner) takes $O(d)$ units, so the maximum memory (outside of Algorithm 13) is $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Opt}} + d)$. The stated results follow from combining the computational complexities analyzed in Lemma 4.15. □

4.7.2 Proofs and Supporting Results for Section 4.4.1

In this section we provide supporting results concerning Algorithm 10 (Section 4.7.2.1), and then give the proof of Theorem 4.12 (Section 4.7.2.2).

4.7.2.1 Supporting Results

Lemma 4.17. *In Algorithm 10 (Eq. (4.6)), there exists a unique choice of $\lambda > 0$ such that $\sum_{a \in \mathcal{A}} p_t(a) = 1$, and its value lies in $[\frac{1}{2}, 1]$.*

Proof of Lemma 4.17. Define $h(\lambda) := \sum_{a \in \mathrm{supp}(q_t)} \frac{q_t(a)}{\lambda + \eta(\widehat{f}_t(x_t, \widehat{a}_t) - \widehat{f}_t(x_t, a))}$. We first no-

tice that $h(\lambda)$ is continuous and strictly decreasing over $(0, \infty)$. We further have

$$h(1/2) \geqslant \frac{q_t(\widehat{a}_t)}{1/2 + \eta(\widehat{f}_t(x_t, \widehat{a}_t) - \widehat{f}_t(x_t, \widehat{a}_t))} \geqslant \frac{1/2}{1/2} = 1;$$

and

$$h(1) \leqslant \sum_{a \in \mathrm{supp}(q_t)} q_t(a) = \frac{1}{2} + \frac{1}{2} \sum_{a \in \mathrm{supp}(q_t^{\mathrm{opt}})} q_t^{\mathrm{opt}}(a) = 1.$$

As a result, there exists a unique normalization constant $\lambda^\star \in [\frac{1}{2}, 1]$ such that $h(\lambda^\star) = 1$. □

Lemma 4.11. *For any $\gamma > 0$, by setting $\eta = \gamma/(C_{\mathrm{opt}} \cdot d)$, the exploration strategy used in Algorithm 10 certifies that $\mathsf{dec}_\gamma(\mathcal{F}) = O(C_{\mathrm{opt}} \cdot d/\gamma)$.*

Proof of Lemma 4.11. As in the proof of Lemma 4.10, we use the linear structure of the action space to rewrite the Decision-Estimation Coefficient $\mathsf{dec}_\gamma(\mathcal{F}; \widehat{f}, x)$ as

$$\mathsf{dec}_\gamma(\mathcal{G}; \widehat{g}, x) :=$$

$$\inf_{p \in \Delta(\mathcal{A})} \sup_{a^\star \in \mathcal{A}} \sup_{g^\star \in \mathcal{G}} \mathbb{E}_{a \sim p}\left[\left\langle \phi(x, a^\star) - \phi(x, a), g^\star(x)\right\rangle - \gamma \cdot \left(\left\langle \phi(x, a), g^\star(x) - \widehat{g}(x)\right\rangle\right)^2\right],$$

Where $\widehat{g}$ is such that $\widehat{f} = f_{\widehat{g}}$. We will bound the quantity above uniformly for all $x \in \mathcal{X}$ and $\widehat{g} : \mathcal{X} \to \mathbb{R}^d$.

Denote $\widehat{a} := \arg\max_{a \in \mathcal{A}} \left\langle \phi(x, a), \widehat{g}(x)\right\rangle$, $a^\star := \arg\max_{a \in \mathcal{A}} \left\langle \phi(x, a), g^\star(x)\right\rangle$ and $q^{\mathrm{opt}} \in \Delta(\mathcal{A})$ be a C_{opt}-approximate optimal design with respect to the reweighted embedding $\overline{\phi}(x, \cdot))$. We use the setting $\eta = \frac{\gamma}{C_{\mathrm{opt}} \cdot d}$ throughout the proof. Recall that for the sampling distribution in Algorithm 10, we set $q := \frac{1}{2} q^{\mathrm{opt}} + \frac{1}{2} \mathbb{I}_{\widehat{a}}$ and define

$$p(a) = \frac{q(a)}{\lambda + \frac{\gamma}{C_{\mathrm{opt}} \cdot d}\left(\left\langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x)\right\rangle\right)}, \tag{4.11}$$

where $\lambda \in [\frac{1}{2}, 1]$ is a normalization constant (cf. Lemma 4.17).

We decompose the regret of the distribution p in Eq. (4.11) as

$$\begin{aligned}
&\mathbb{E}_{a\sim p}\Big[\big\langle \phi(x,a^\star)-\phi(x,a), g^\star(x)\big\rangle\Big] \\
&= \mathbb{E}_{a\sim p}\Big[\big\langle \phi(x,\widehat{a})-\phi(x,a), \widehat{g}(x)\big\rangle\Big] + \mathbb{E}_{a\sim p}\Big[\big\langle \phi(x,a), \widehat{g}(x)-g^\star(x)\big\rangle\Big] \\
&\quad + \big\langle \phi(x,a^\star), g^\star(x)-\widehat{g}(x)\big\rangle + \big\langle \phi(x,a^\star)-\phi(x,\widehat{a}), \widehat{g}(x)\big\rangle. \qquad (4.12)
\end{aligned}$$

Writing out the expectation, the first term in Eq. (4.12) is upper bounded as follows.

$$\begin{aligned}
&\mathbb{E}_{a\sim p}\Big[\big\langle \phi(x,\widehat{a})-\phi(x,a), \widehat{g}(x)\big\rangle\Big] \\
&= \sum_{a\in \mathrm{supp}(q^{\mathrm{opt}})\cup\{\widehat{a}\}} p(a)\cdot\big\langle \phi(x,\widehat{a})-\phi(x,a), \widehat{g}(x)\big\rangle \\
&< \sum_{a\in \mathrm{supp}(q^{\mathrm{opt}})} \frac{q^{\mathrm{opt}}(a)/2}{\frac{\gamma}{C_{\mathrm{opt}}\cdot d}\big(\big\langle \phi(x,\widehat{a})-\phi(x,a), \widehat{g}(x)\big\rangle\big)}\cdot\big\langle \phi(x,\widehat{a})-\phi(x,a), \widehat{g}(x)\big\rangle \\
&\leqslant \frac{C_{\mathrm{opt}}\cdot d}{2\gamma},
\end{aligned}$$

where we use that $\lambda > 0$ in the second inequality (with the convention that $\frac{0}{0}=0$).

The second term in Eq. (4.12) can be upper bounded as in the proof of Lemma 4.10, by applying the AM-GM inequality:

$$\mathbb{E}_{a\sim p}\Big[\big\langle \phi(x,a), g^\star(x)-\widehat{g}(x)\big\rangle\Big] \leqslant \frac{\gamma}{2}\cdot\mathbb{E}_{a\sim p}\Big[\big(\big\langle \phi(x,a), \widehat{g}(x)-g^\star(x)\big\rangle\big)^2\Big] + \frac{1}{2\gamma}.$$

The third term in Eq. (4.12) is the most involved. To begin, we define $V(p) := \mathbb{E}_{a\sim p}[\phi(x,a)\phi(x,a)^\top]$ and apply the following standard bound:

$$\begin{aligned}
\big\langle \phi(x,a^\star), \widehat{g}(x)-g^\star(x)\big\rangle &\leqslant \big\|\phi(x,a^\star)\big\|_{V(p)^{-1}}\cdot\big\|g^\star(x)-\widehat{g}(x)\big\|_{V(p)} \\
&\leqslant \frac{1}{2\gamma}\cdot\big\|\phi(x,a^\star)\big\|^2_{V(p)^{-1}} + \frac{\gamma}{2}\cdot\big\|g^\star(x)-\widehat{g}(x)\big\|^2_{V(p)} \\
&= \frac{1}{2\gamma}\cdot\big\|\phi(x,a^\star)\big\|^2_{V(p)^{-1}} + \frac{\gamma}{2}\cdot\mathbb{E}_{a\sim p}\Big[\big(\phi(x,a), g^\star(x)-\widehat{g}(x)\big)^2\Big], \qquad (4.13)
\end{aligned}$$

where the second line follows from the AM-GM inequality. The second term in Eq. (4.13) matches the bound we desired, so it remains to bound the first term. Let $\check{q}^{\mathrm{opt}}$ be the following sub-probability measure:

$$\check{q}^{\mathrm{opt}}(a) := \frac{q^{\mathrm{opt}}(a)/2}{\lambda + \frac{\gamma}{C_{\mathrm{opt}} \cdot d}\left(\left\langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x)\right\rangle\right)},$$

and let $V(\check{q}^{\mathrm{opt}}) := \mathbb{E}_{a \sim \check{q}^{\mathrm{opt}}}[\phi(x, a)\phi(x, a)^\top]$. We clearly have $V(p) \succeq V(\check{q}^{\mathrm{opt}})$ from the definition of p (cf. Eq. (4.11)). We observe that

$$\begin{aligned}
V(\check{q}^{\mathrm{opt}}) &= \sum_{a \in \mathrm{supp}(\check{q}^{\mathrm{opt}})} \check{q}^{\mathrm{opt}}(a)\phi(x, a)\phi(x, a)^\top \\
&= \frac{1}{2} \cdot \sum_{a \in \mathrm{supp}(q^{\mathrm{opt}})} q^{\mathrm{opt}}(a)\overline{\phi}(x, a)\overline{\phi}(x, a)^\top \cdot \frac{1 + \frac{\gamma}{C_{\mathrm{opt}} \cdot d}\left(\left\langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x)\right\rangle\right)}{\lambda + \frac{\gamma}{C_{\mathrm{opt}} \cdot d}\left(\left\langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x)\right\rangle\right)} \\
&\succeq \frac{1}{2} \cdot \sum_{a \in \mathrm{supp}(q^{\mathrm{opt}})} q^{\mathrm{opt}}(a)\overline{\phi}(x, a)\overline{\phi}(x, a)^\top =: \frac{1}{2}\overline{V}(q^{\mathrm{opt}}),
\end{aligned}$$

where the last line uses that $\lambda \leqslant 1$. Since $\overline{V}(q^{\mathrm{opt}})$ is positive-definite by construction, we have that $V(p)^{-1} \preceq V(\check{q}^{\mathrm{opt}})^{-1} \preceq 2 \cdot \overline{V}(q^{\mathrm{opt}})^{-1}$. As a result,

$$\begin{aligned}
\frac{1}{2\gamma} \cdot \left\|\phi(x, a^\star)\right\|^2_{V(p)^{-1}} &\leqslant \frac{1}{\gamma} \cdot \left\|\phi(x, a^\star)\right\|^2_{\overline{V}(q^{\mathrm{opt}})^{-1}} \\
&= \frac{1 + \frac{\gamma}{C_{\mathrm{opt}} \cdot d}\left(\left\langle \phi(x, \widehat{a}) - \phi(x, a^\star), \widehat{g}(x)\right\rangle\right)}{\gamma} \cdot \left\|\overline{\phi}(x, a^\star)\right\|^2_{\overline{V}(q^{\mathrm{opt}})^{-1}} \\
&\leqslant \frac{C_{\mathrm{opt}} \cdot d}{\gamma} + \left\langle \phi(x, \widehat{a}) - \phi(x, a^\star), \widehat{g}(x)\right\rangle, \qquad (4.14)
\end{aligned}$$

where the last line uses that $\left\|\overline{\phi}(x, a^\star)\right\|^2_{\overline{V}(q^{\mathrm{opt}})^{-1}} \leqslant C_{\mathrm{opt}} \cdot d$, since q^{opt} is a C_{opt}-approximate optimal design for the set $\left\{\overline{\phi}(x, a)\right\}_{a \in \mathcal{A}}$. Finally, we observe that the second term in Eq. (4.14) is cancelled out by the forth term in Eq. (4.12).

Summarizing the bounds on the terms in Eq. (4.12) leads to:

$$\mathsf{dec}_\gamma(\mathcal{F}) = \inf_{p\in\Delta(\mathcal{A})} \sup_{a^\star\in\mathcal{A}} \sup_{g^\star\in\mathcal{G}} \mathsf{dec}_\gamma(\mathcal{G}; \widehat{g}, x) \leqslant \frac{C_{\mathrm{opt}}\cdot d}{2\gamma} + \frac{1}{2\gamma} + \frac{C_{\mathrm{opt}}\cdot d}{\gamma} \leqslant \frac{2\,C_{\mathrm{opt}}\cdot d}{\gamma}.$$

□

4.7.2.2 Proof of Theorem 4.12

Theorem 4.12. *Let $\delta \in (0,1)$ be given. With a C_{opt}-approximate optimal design subroutine and an appropriate choice for $\gamma > 0$, Algorithm 10 ensures that with probability at least $1-\delta$,*

$$\mathbf{Reg}_{\mathsf{CB}}(T) = O\Big(\sqrt{C_{\mathrm{opt}}\cdot d\,T\left(\mathbf{Reg}_{\mathsf{Sq}}(T) + \log(\delta^{-1})\right)}\Big).$$

In particular, when invoked with Algorithm 11 (with $C = 2$) as a subroutine, the algorithm has

$$\mathbf{Reg}_{\mathsf{CB}}(T) = O\Big(d\sqrt{T\left(\mathbf{Reg}_{\mathsf{Sq}}(T) + \log(\delta^{-1})\right)}\Big),$$

and has per-round runtime $O(\mathcal{T}_{\mathsf{Sq}} + (\mathcal{T}_{\mathsf{Opt}}\cdot d^3 + d^4)\cdot\log^2\left(\frac{T}{r}\right))$ and the maximum memory $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Opt}} + d^2 + d\log\left(\frac{T}{r}\right))$.

Proof. Combining Lemma 4.11 with Lemma 4.16, we have

$$\mathbf{Reg}_{\mathsf{CB}}(T) \leqslant 2T\cdot\frac{C_{\mathrm{opt}}\cdot d}{\gamma} + 2\gamma\cdot\mathbf{Reg}_{\mathsf{Sq}}(T) + 64\gamma\cdot\log(2\delta^{-1}) + \sqrt{8T\log(2\delta^{-1})}.$$

The theorem follows by choosing

$$\gamma = \left(\frac{C_{\mathrm{opt}}\cdot d\,T}{\mathbf{Reg}_{\mathsf{Sq}}(T) + 32\log(2\delta^{-1})}\right)^{1/2}.$$

In particular, when Algorithm 11 is invoked as the subroutine with parameter $C = 2$, we may take $C_{\mathrm{opt}} = 4d$.

Computational complexity. We now discuss the per-round computational complexity of Algorithm 10. We analyze a variant of the sampling rule specified in Section 4.7.4.2 that does not require computation of the normalization constant. Outside of the runtime and memory requirements required to compute the barycentric spanner using Algorithm 11, which are stated in Theorem 4.13, Algorithm 10 uses $O(1)$ calls to the oracle $\mathbf{Alg}_{\mathsf{Sq}}$ to obtain $\widehat{g}_t(x_t) \in \mathbb{R}^d$ and update $\widehat{f}_t$, and uses a single call to $\mathbf{Alg}_{\mathsf{Opt}}$ to compute $\widehat{a}_t$. With $\widehat{g}_t(x_t)$ and $\widehat{a}_t$, we can compute $\widehat{f}_t(x_t, \widehat{a}_t) - \widehat{f}_t(x_t, a) = \langle \phi(x_t, \widehat{a}_t) - \phi(x_t, a), \widehat{g}_t(x_t) \rangle$ in $O(d)$ time for any $a \in \mathcal{A}$; thus, with the optimal design q_t^{opt} returned by Algorithm 11 (represented as a barycentric spanner), we can construct the sampling distribution p_t in $O(d^2)$ time. Sampling from p_t takes $O(d)$ time since $|\mathrm{supp}(p_t)| \leqslant d+1$. This adds up to runtime $O(\mathcal{T}_{\mathsf{Sq}}+\mathcal{T}_{\mathsf{Opt}}+d^2)$. In terms of memory, calling $\mathbf{Alg}_{\mathsf{Sq}}$ and $\mathbf{Alg}_{\mathsf{Opt}}$ takes $O(\mathcal{M}_{\mathsf{Sq}}+\mathcal{M}_{\mathsf{Opt}})$ units, and maintaining the distribution p_t (the barycentric spanner) takes $O(d)$ units, so the maximum memory (outside of Algorithm 11) is $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Opt}} + d)$. The stated results follow from combining the computational complexities analyzed in Theorem 4.13 , together with the choice of γ described above. □

4.7.3 Proofs and Supporting Results for Section 4.4.2

This section of the appendix is dedicated to the analysis of Algorithm 11, and organized as follows.

- First, in Section 4.7.3.1, we analyze Algorithm 12, a subroutine of Algorithm 11 which implements a linear optimization oracle for the reweighted action set used in the algorithm.
- Next, in Section 4.7.3.2, we prove Theorem 4.13, the main theorem concerning the performance of Algorithm 11.
- Finally, in Section 4.7.3.3, we discuss settings in which the initialization step required by Algorithm 11 can be performed efficiently.

Throughout this section of the appendix, we assume that the context $x \in \mathcal{X}$ and estimator $\widehat{g} : \mathcal{X} \to \mathbb{R}^d$—which are arguments to Algorithm 11 and Algorithm 12—are fixed.

4.7.3.1 Analysis of Algorithm 12 (Linear Optimization Oracle for Reweighted Embeddings)

A first step is to construct an (approximate) argmax oracle (after taking absolute value) with respect to the reweighted embedding $\overline{\phi}$. Recall that the goal of Algorithm 12 is to implement a linear optimization oracle for the reweighted embeddings constructed by Algorithm 11. That is, for any $\theta \in \mathbb{R}^d$, we would like to compute an action that (approximately) solves

$$\arg\max_{a \in \mathcal{A}} \left|\left\langle \overline{\phi}(x, a), \theta \right\rangle\right| = \arg\max_{a \in \mathcal{A}} \left\langle \overline{\phi}(x, a), \theta \right\rangle^2.$$

Define

$$\iota(a) := \langle \overline{\phi}(x, a), \theta \rangle^2, \quad \text{and} \quad a^\star := \arg\max_{a \in \mathcal{A}} \iota(a). \tag{4.15}$$

The main result of this section, Theorem 4.18, shows that Algorithm 12 identifies an action that achieves the maximum value in Eq. (4.15) up to a multiplicative constant.

Theorem 4.18. *Fix any $\eta > 0$, $r \in (0, 1)$. Suppose $\zeta \leqslant \sqrt{\iota(a^\star)} \leqslant 1$ for some $\zeta > 0$. Then Algorithm 12 identifies an action $\check{a}$ such that $\sqrt{\iota(\check{a})} \geqslant \frac{\sqrt{2}}{2} \cdot \sqrt{\iota(a^\star)}$, and does so with runtime $O((\mathcal{T}_{\mathsf{Opt}} + d) \cdot \log(e \vee \frac{\eta}{\zeta}))$ and maximum memory $O(\mathcal{M}_{\mathsf{Opt}} + \log(e \vee \frac{\eta}{\zeta}) + d)$.*

Proof of Theorem 4.18. Recall from Eq. (4.5) that we have

$$\left\langle \overline{\phi}(x, a), \theta \right\rangle^2 = \left(\frac{\left\langle \phi(x, a), \theta \right\rangle}{\sqrt{1 + \eta \left\langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x) \right\rangle}} \right)^2 = \frac{\left\langle \phi(x, a), \theta \right\rangle^2}{1 + \eta \left\langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x) \right\rangle},$$

where $\widehat{a} := \arg\max_{a \in \mathcal{A}} \langle \phi(x, a), \widehat{g}(x) \rangle$; note that the denominator is at least 1. To

proceed, we use that for any $X \in \mathbb{R}$ and $Y^2 > 0$, we have

$$\frac{X^2}{Y^2} = \sup_{\varepsilon \in \mathbb{R}} \{2\varepsilon X - \varepsilon^2 Y^2\}.$$

Taking $X = \langle \phi(x, a), \theta \rangle$ and $Y^2 = 1 + \eta \langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x) \rangle$ above, we can write

$$\begin{aligned}
\langle \overline{\phi}(x, a), \theta \rangle^2 &= \frac{\langle \phi(x, a), \theta \rangle^2}{1 + \eta \langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x) \rangle} \\
&= \sup_{\varepsilon \in \mathbb{R}} \Big\{ 2\varepsilon \langle \phi(x, a), \theta \rangle - \varepsilon^2 \cdot \big(1 + \eta \langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x) \rangle\big) \Big\} \quad (4.16) \\
&= \sup_{\varepsilon \in \mathbb{R}} \Big\{ \langle \phi(x, a), 2\varepsilon\theta + \eta\varepsilon^2 \widehat{g}(x) \rangle - \varepsilon^2 - \eta\varepsilon^2 \langle \phi(x, \widehat{a}), \widehat{g}(x) \rangle \Big\}. \quad (4.17)
\end{aligned}$$

The key property of this representation is that for any *fixed* $\varepsilon \in \mathbb{R}$, Eq. (4.17) is a linear function of the *unweighted* embedding ϕ, and hence can be optimized using $\mathbf{Alg}_{\mathsf{Opt}}$. In particular, for any fixed $\varepsilon \in \mathbb{R}$, consider the following linear optimization problem, which can be solved by calling $\mathbf{Alg}_{\mathsf{Opt}}$:

$$\arg\max_{a \in \mathcal{A}} \Big\{ 2\varepsilon \langle \phi(x, a), \theta \rangle - \varepsilon^2 \cdot \big(1 + \eta \langle \phi(x, \widehat{a}) - \phi(x, a), \widehat{g}(x) \rangle\big) \Big\} =: \arg\max_{a \in \mathcal{A}} W(a; \varepsilon). \tag{4.18}$$

Define

$$\varepsilon^\star := \frac{\langle \phi(x, a^\star), \theta \rangle}{1 + \eta \langle \phi(x, \widehat{a}) - \phi(x, a^\star), \widehat{g}(x) \rangle}. \tag{4.19}$$

If $\varepsilon^\star$ was known (which is not the case, since $a^\star$ is unknown), we could set $\varepsilon = \varepsilon^\star$ in Eq. (4.18) and compute an action $\bar{a} := \arg\max_{a \in \mathcal{A}} W(a; \varepsilon^\star)$ using a single oracle call. We would then have $\iota(\bar{a}) \geqslant W(\bar{a}; \varepsilon^\star) \geqslant W(a^\star; \varepsilon^\star) = \iota(a^\star)$, which follows because $\varepsilon^\star$ is the maximizer in Eq. (4.16) for $a = a^\star$.

To get around the fact that $\varepsilon^\star$ is unknown, Algorithm 12 performs a grid search over possible values of ε. To show that the procedure succeeds, we begin by

bounding the range of $\varepsilon^\star$. With some rewriting, we have

$$|\varepsilon^\star| = \frac{\sqrt{\iota(a^\star)}}{\sqrt{1+\eta\langle\phi(x,\widehat{a})-\phi(x,a^\star),\widehat{g}(x)\rangle}}.$$

Since $0 < \zeta \leqslant \sqrt{\iota(a^\star)} \leqslant 1$, we have

$$\bar{\zeta} := \frac{\zeta}{\sqrt{1+2\eta}} \leqslant |\varepsilon^\star| \leqslant 1.$$

Algorithm 12 performs a $(3/4)$-multiplicative grid search over the intervals $[\bar{\zeta}, 1]$ and $[-1, -\bar{\zeta}]$, which uses $2\lceil \log_{\frac{4}{3}}(\bar{\zeta}^{-1})\rceil = O(\log(e \vee \frac{\eta}{\zeta}))$ grid points. It is immediate to that the grid contains $\bar{\varepsilon} \in \mathbb{R}$ such that $\bar{\varepsilon}\cdot\varepsilon^\star > 0$ and $\frac{3}{4}|\varepsilon^\star| \leqslant |\bar{\varepsilon}| \leqslant |\varepsilon^\star|$. Invoking Lemma 4.19 (stated and proven in the sequel) with $\bar{a} := \arg\max_{a\in\mathcal{A}} W(a;\bar{\varepsilon})$ implies that $\iota(\bar{a}) \geqslant \frac{1}{2}\iota(a^\star)$. To conclude, recall that Algorithm 12 outputs the maximizer

$$\check{a} := \arg\max_{a\in\widehat{\mathcal{A}}} \iota(a),$$

where $\widehat{\mathcal{A}}$ is the set of argmax actions encountered by the grid search. Since $\bar{a} \in \widehat{\mathcal{A}}$, we have $\iota(\check{a}) \geqslant \iota(\bar{a}) \geqslant \frac{1}{2}\iota(a^\star)$ as desired.

Computational complexity. Finally, we bound the computational complexity of Algorithm 12. Algorithm 12 maintains a grid of $O(\log(e\vee\frac{\eta}{\zeta}))$ points, and hence calls the oracle $\mathbf{Alg}_{\mathsf{Opt}}$ $O(\log(e\vee\frac{\eta}{\zeta}))$ in total; this takes $O(\mathcal{T}_{\mathsf{Opt}}\cdot\log(e\vee\frac{\eta}{\zeta}))$ time. Computing the final maximizer from the set $\widehat{\mathcal{A}}$, which contains $O(\log(e\vee\frac{\eta}{\zeta}))$ actions, takes $O(d\log(e\vee\frac{\eta}{\zeta}))$ time (compute each $\langle\overline{\phi}(x,a),\bar{\theta}\rangle^2$ takes $O(d)$ time). Hence, the total runtime of Algorithm 12 adds up to $O((\mathcal{T}_{\mathsf{Opt}}+d)\cdot\log(e\vee\frac{\eta}{\zeta}))$. The maximum memory requirement is $O(\mathcal{M}_{\mathsf{Opt}}+\log(e\vee\frac{\eta}{\zeta})+d)$, follows from calling $\mathbf{Alg}_{\mathsf{Opt}}$, and storing $\mathcal{E}, \widehat{\mathcal{A}}$ and other terms such as $\widehat{g}(x), \theta, \bar{\theta}, \phi(x,a), \overline{\phi}(x,a)$. □

Supporting Results.

Lemma 4.19. *Let $\varepsilon^\star$ be defined as in Eq. (4.19). Suppose $\bar{\varepsilon} \in \mathbb{R}$ has $\bar{\varepsilon}\cdot\varepsilon^\star > 0$ and $\frac{3}{4}|\varepsilon^\star| \leqslant |\bar{\varepsilon}| \leqslant |\varepsilon^\star|$. Then, if $\bar{a} := \arg\max_{a\in\mathcal{A}} W(a;\bar{\varepsilon})$, we have $\iota(\bar{a}) \geqslant \frac{1}{2}\iota(a^\star)$.*

Proof of Lemma 4.19. First observe that using the definition of $\iota(a)$, along with Eq. (4.16) and Eq. (4.18), we have $\iota(\bar{a}) \geqslant W(\bar{a}; \bar{\varepsilon}) \geqslant W(a^\star; \bar{\varepsilon})$, where the second inequality uses that $\bar{a} := \arg\max_{a \in \mathcal{A}} W(a; \bar{\varepsilon})$. Since $\bar{\varepsilon} \cdot \varepsilon^\star > 0$, we have $\operatorname{sign}(\bar{\varepsilon} \cdot \langle \phi(x, a^\star), \theta \rangle) = \operatorname{sign}(\varepsilon^\star \cdot \langle \phi(x, a^\star), \theta \rangle)$. If $\operatorname{sign}(\bar{\varepsilon} \cdot \langle \phi(x, a^\star), \theta \rangle) \geqslant 0$, then since $\frac{3}{4}|\varepsilon^\star| \leqslant |\bar{\varepsilon}| \leqslant |\varepsilon^\star|$, we have

$$\begin{aligned} W(a^\star; \bar{\varepsilon}) &= 2\bar{\varepsilon} \cdot \big\langle \phi(x, a^\star), \theta \big\rangle - \bar{\varepsilon}^2 \cdot \big(1 + \eta \big\langle \phi(x, \widehat{a}) - \phi(x, a^\star), \widehat{g}(x) \big\rangle\big) \\ &\geqslant \frac{3}{2}\varepsilon^\star \cdot \big\langle \phi(x, a^\star), \theta \big\rangle - (\varepsilon^\star)^2 \cdot \big(1 + \eta \big\langle \phi(x, \widehat{a}) - \phi(x, a^\star), \widehat{g}(x) \big\rangle\big) \\ &= \frac{1}{2} \frac{\langle \phi(x, a^\star), \theta \rangle^2}{1 + \eta \langle \phi(x, \widehat{a}) - \phi(x, a^\star), \widehat{g}(x) \rangle} = \frac{1}{2}\iota(a^\star), \end{aligned}$$

where we use that $1 + \eta \big\langle \phi(x, \widehat{a}) - \phi(x, a^\star), \widehat{g}(x) \big\rangle \geqslant 1$ for the first inequality and use the definition of $\varepsilon^\star$ for the second equality.

On the other hand, when $\operatorname{sign}(\bar{\varepsilon} \cdot \langle \phi(x, a^\star), \theta \rangle) < 0$, we similarly have

$$\begin{aligned} W(a^\star; \bar{\varepsilon}) &= 2\bar{\varepsilon} \cdot \big\langle \phi(x, a^\star), \theta \big\rangle - \bar{\varepsilon}^2 \cdot \big(1 + \eta \big\langle \phi(x, \widehat{a}) - \phi(x, a^\star), \widehat{g}(x) \big\rangle\big) \\ &\geqslant 2\varepsilon^\star \cdot \big\langle \phi(x, a^\star), \theta \big\rangle - (\varepsilon^\star)^2 \cdot \big(1 + \eta \big\langle \phi(x, \widehat{a}) - \phi(x, a^\star), \widehat{g}(x) \big\rangle\big) = \iota(a^\star). \end{aligned}$$

Summarizing both cases, we have $\iota(\bar{a}) \geqslant \frac{1}{2}\iota(a^\star)$. □

4.7.3.2 Proof of Theorem 4.13

Theorem 4.13. *Suppose that Algorithm 11 is invoked with parameters $\eta > 0$, $r \in (0, 1)$, and $C > \sqrt{2}$, and that the initialization set S satisfies $|\det(\phi(x, S))| \geqslant r^d$. Then the algorithm returns a C-approximate barycentric spanner with respect to the reweighted embedding set $\{\overline{\phi}(x, a)\}_{a \in \mathcal{A}}$, and does so with $O((\mathcal{T}_{\mathsf{Opt}} \cdot d^3 + d^4) \cdot \log^2(e \vee \frac{\eta}{r}))$ runtime and $O(\mathcal{M}_{\mathsf{Opt}} + d^2 + d \log(e \vee \frac{\eta}{r}))$ memory.*

Proof of Theorem 4.13. We begin by examining the range of $\sqrt{\iota(a^\star)}$ used in Theorem 4.18. Note that the linear function θ passed as an argument to Algorithm 11 takes the form $\overline{\phi}(x, a) \mapsto \det(\overline{\phi}(x, S_i(a)))$, i.e., $\langle \overline{\phi}(x, a), \theta \rangle = \det(\overline{\phi}(x, S_i(a)))$,

where $\mathcal{S}_i(a) := (a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_d)$. For the upper bound, we have

$$|\langle \overline{\phi}(x, a^\star), \theta \rangle| = |\det(\overline{\phi}(x, \mathcal{S}_i(a^\star)))| \leqslant \prod_{a \in \mathcal{S}_i(a^\star)} \|\overline{\phi}(x, a)\|_2^d \leqslant \sup_{a \in \mathcal{A}} \|\phi(x, a)\|_2^d \leqslant 1$$

by Hadamard's inequality and the fact that the reweighting appearing in Eq. (4.5) enjoys $\|\overline{\phi}(x, a)\|_2 \leqslant \|\phi(x, a)\|_2$. This shows that $\sqrt{\iota(a^\star)} \leqslant 1$. For the lower bound, we first recall that in Algorithm 11, the set $\mathcal{S}$ is initialized to have $|\det(\phi(x, \mathcal{S}))| \geqslant r^d$, and thus $|\det(\overline{\phi}(x, \mathcal{S}))| \geqslant \bar{r}^d$, where $\bar{r} := \frac{r}{\sqrt{1+2\eta}}$ accounts for the reweighting in Eq. (4.5). Next, we observe that as a consequence of the update rule in Algorithm 11, we are guaranteed that $|\det(\overline{\phi}(x, \mathcal{S}))| \geqslant \bar{r}^d$ across all rounds. Thus, whenever Algorithm 12 is invoked with the linear function θ described above, there must exist an action $a \in \mathcal{A}$ such that $|\langle \overline{\phi}(x, a), \theta \rangle| \geqslant \bar{r}^d$, which implies that $\sqrt{\iota(a^\star)} \geqslant \bar{r}^d$ and we can take $\zeta := \bar{r}^d$ in Theorem 4.18.

We next bound the number of iterations of the while-loop before the algorithm terminates. Let $\overline{C} := \frac{\sqrt{2}}{2} \cdot C > 1$. At each iteration (beginning from line 3) of Algorithm 11, one of two outcomes occurs:

1. We find an index $i \in [d]$ and an action $a \in \mathcal{A}$ such that $|\det(\overline{\phi}(x, \mathcal{S}_i(a)))| > \overline{C}|\det(\overline{\phi}(x, \mathcal{S}))|$, and update $a_i = a$.

2. We conclude that $\sup_{a \in \mathcal{A}} \max_{i \in [d]} |\det(\overline{\phi}(x, \mathcal{S}_i(a)))| \leqslant C|\det(\overline{\phi}(x, \mathcal{S}))|$ and terminate the algorithm.

We observe that (i) the initial set $\mathcal{S}$ has $|\det(\overline{\phi}(x, \mathcal{S}))| \geqslant \bar{r}^d$ with $\bar{r} := \frac{r}{\sqrt{1+2\eta}}$ (as discussed before), (ii) $\sup_{\mathcal{S} \subseteq \mathcal{A}, |\mathcal{S}| = d} |\det(\overline{\phi}(x, \mathcal{S}))| \leqslant 1$ by Hadamard's inequality, and (iii) each update of $\mathcal{S}$ increases the (absolute) determinant by a factor of $\overline{C}$. Thus, fix any $C > \sqrt{2}$, we are guaranteed that Algorithm 11 terminates within $O(d \log(e \vee \frac{\eta}{r}))$ iterations of the while-loop.

We now discuss the correctness of Algorithm 11, i.e., when terminated, the set $\mathcal{S}$ is a C-approximate barycentric spanner with respect to the reweighted embedding $\overline{\phi}$. First, note that by Theorem 4.18, Algorithm 12 is guaranteed to identify an action $\check{a} \in \mathcal{A}$ such that $|\det(\overline{\phi}(x, \mathcal{S}_i(\check{a})))| > \overline{C}|\det(\overline{\phi}(x, \mathcal{S}))|$ as long as there exists an action

$a^\star \in \mathcal{A}$ such that $|\det(\overline{\phi}(x, \mathcal{S}_i(a^\star)))| > C|\det(\overline{\phi}(x, \mathcal{S}))|$. As a result, by Observation 2.3 in Awerbuch and Kleinberg (2008), if no update is made and Algorithm 11 terminates, we have identified a C-approximate barycentric spanner with respect to embedding $\overline{\phi}$.

Computational complexity. We provide the computational complexity analysis for Algorithm 11 in the following. We use $\overline{\Phi}_{\mathcal{S}}$ to denote the matrix whose k-th column is $\overline{\phi}(x, a_k)$ with $a_k \in \mathcal{S}$.

- *Initialization.* We first notice that, given $\widehat{g}(x) \in \mathbb{R}^d$ and $\widehat{a} := \arg\max_{a \in \mathcal{A}} \langle \phi(x, a), \widehat{g}(x) \rangle$, it takes $O(d)$ time to compute $\overline{\phi}(x, a)$ for any $a \in \mathcal{A}$. Thus, computing $\det(\overline{\Phi}_{\mathcal{S}})$ and $\overline{\Phi}_{\mathcal{S}}^{-1}$ takes $O(d^2 + d^\omega) = O(d^\omega)$ time, where we use $O(d^\omega)$ (with $2 \leqslant \omega \leqslant 3$) to denote the time of computing matrix determinant/inversion. The maximum memory requirement is $O(d^2)$, following from the storage of $\{\overline{\phi}(x, a)\}_{a \in \mathcal{S}}$ and $\overline{\Phi}_{\mathcal{S}}^{-1}$.

- *Outer loops (lines 1-2).* We have already shown that Algorithm 13 terminates within $O(d \log(e \vee \frac{\eta}{r}))$ iterations of the while-loop (line 2). It is also clear that the for-loop (line 2) is invoked at most d times.

- *Computational complexity for lines 3-7.* We discuss how to efficiently implement this part using rank-one updates. We analyze the computational complexity for each line in the following. The analysis largely follows from the proof of Lemma 4.15.

  - *Line 3.* Using rank-one update of the matrix determinant (as discussed in the proof of Lemma 4.15), we have

    $$\det(\overline{\phi}(x, a_1), \ldots, \overline{\phi}(x, a_{i-1}), Y, \overline{\phi}(x, a_{i+1}), \ldots, \overline{\phi}(x, a_d)) = \langle Y, \theta \rangle,$$

    where $\theta = \det(\overline{\Phi}_{\mathcal{S}}) \cdot (\overline{\Phi}_{\mathcal{S}}^{-1})^\top e_i$. Thus, whenever $\det(\overline{\Phi}_{\mathcal{S}})$ and $\overline{\Phi}_{\mathcal{S}}^{-1}$ are known, compute θ takes $O(d)$ time. The maximum memory requirement is $O(d^2)$, following from the storage of $\overline{\Phi}_{\mathcal{S}}^{-1}$.

- *Line 4.* When θ is computed, we can compute a by invoking IGW-ArgMax (Algorithm 12). As discussed in Theorem 4.18, this step takes runtime $O((\mathcal{T}_{\mathsf{Opt}} \cdot d + d^2) \cdot \log(e \vee \frac{\eta}{r}))$ and maximum memory $O(\mathcal{M}_{\mathsf{Opt}} + d \log(e \vee \frac{\eta}{r}) + d)$ (by taking $\zeta = \bar{r}^d$ as discussed before).
- *Line 5.* Once θ and $\det(\overline{\Phi}_S)$ are computed, checking the updating criteria takes $O(d)$ time. The maximum memory requirement is $O(d)$, following from the storage of $\overline{\phi}(x, a)$ and θ.
- *Line 6.* As discussed in the proof of Lemma 4.15, if an update $a_i = a$ is made, we can update $\det(\overline{\Phi}_S)$ and $\overline{\Phi}_S^{-1}$ using rank-one updates with $O(d^2)$ time and memory.

Thus, using rank-one updates, the total runtime for line 3-7 adds up to $O((\mathcal{T}_{\mathsf{Opt}} \cdot d + d^2) \cdot \log(e \vee \frac{\eta}{r}))$ and maximum memory requirement is $O(\mathcal{M}_{\mathsf{Opt}} + d^2 + d \log(e \vee \frac{\eta}{r}))$.

To summarize, Algorithm 13 has runtime $O((\mathcal{T}_{\mathsf{Opt}} \cdot d^3 + d^4) \cdot \log^2(e \vee \frac{\eta}{r}))$ and uses at most $O(\mathcal{M}_{\mathsf{Opt}} + d^2 + d \log(e \vee \frac{\eta}{r}))$ units of memory. □

4.7.3.3 Efficient Initializations for Algorithm 11

In this section we discuss specific settings in which the initialization required by Algorithm 11 can be computed efficiently. For the first result, we let $\mathsf{Ball}(0, r) := \{x \in \mathbb{R}^d \mid \|x\|_2 \leqslant r\}$ denote the ball of radius r in $\mathbb{R}^d$.

Example 4.20. *Suppose that there exists* $r \in (0, 1)$ *such that* $\mathsf{Ball}(0, r) \subseteq \{\phi(x, a) : a \in \mathcal{A}\}$. *Then by choosing* $S := \{re_1, \ldots, re_d\} \subseteq \mathcal{A}$, *we have* $|\det(\phi(S))| = r^d$.

The next example is stronger, and shows that we can efficiently compute a set with large determinant whenever such a set exists.

Example 4.21. *Suppose there exists a set* $S^\star \subseteq \mathcal{A}$ *such that* $|\det(\phi(S^\star))| \geqslant \bar{r}^d$ *for some* $\bar{r} > 0$. *Then there exists an efficient algorithm that identifies a set* $S \subseteq \mathcal{A}$ *with* $|\det(\phi(S))| \geqslant r^d$ *for* $r := \frac{\bar{r}}{8d}$, *and does so with runtime* $O(\mathcal{T}_{\mathsf{Opt}} \cdot d^2 \log d + d^4 \log d)$ *and memory* $O(\mathcal{M}_{\mathsf{Opt}} + d^2)$.

Proof for Example 4.21. The guarantee is achieved by running Algorithm 13 with $C = 2$. One can show that this strategy achieves the desired approximation guarantee by slightly generalizing the proof of a similar result in Mahabadi et al. (2019). In more detail, Mahabadi et al. (2019) study the problem of identifying a subset $S \subseteq \mathcal{A}$ such that $|S| = k$ and $\det(\Phi_S^\top \Phi_S)$ is (approximately) maximized, where $\Phi_S \in \mathbb{R}^{d \times |S|}$ denotes the matrix whose columns are $\phi(x, a)$ for $a \in S$. We consider the case when $k = d$, and make the following observations.

- We have $\det(\Phi_S^\top \Phi_S) = (\det(\Phi_S))^2 = (\det(\phi(x, S)))^2$. Thus, maximizing $\det(\Phi_S^\top \Phi_S)$ is equivalent to maximizing $|\det(\phi(x, S))|$.
- The Local Search Algorithm provided in Mahabadi et al. (2019) (Algorithm 4.1 therein) has the same update and termination condition as Algorithm 13. As a result, one can show that the conclusion of their Lemma 4.1 also applies to Algorithm 13.

□

4.7.4 Other Details for Experiments

4.7.4.1 Basic Details

Datasets. oneshotwiki (Singh et al., 2012; Vasnetsov, 2018) is a named-entity recognition task where contexts are text phrases preceding and following the mention text, and where actions are text phrases corresponding to the concept names. We use the python package sentence transformers (Reimers and Gurevych, 2019) to separately embed the text preceding and following the reference into $\mathbb{R}^{768}$, and then concatenate, resulting in a context embedding in $\mathbb{R}^{1536}$. We embed the action (mentioned entity) text into $\mathbb{R}^{768}$ and then use SVD on the collection of embedded actions to reduce the dimensionality to $\mathbb{R}^{50}$. The reward function is an indicator function for whether the action corresponds to the actual entity mentioned. oneshotwiki-311 (resp. oneshotwiki-14031) is a subset of this dataset obtained by taking all actions with at least 2000 (resp. 200) examples.

amazon-3m (Bhatia et al., 2016) is an extreme multi-label dataset whose contexts are text phrases corresponding to the title and description of an item, and whose actions are integers corresponding to item tags. We separately embed the title and description phrases using sentence transformers, which leads to a context embedding in $\mathbb{R}^{1536}$. Following the protocol used in Sen et al. (2021), the first 50000 examples are fully supervised, and subsequent examples have bandit feedback. We use Hellinger PCA (Lebret and Collobert, 2014) on the supervised data label cooccurrences to construct the action embeddings in $\mathbb{R}^{800}$. Rewards are binary, and indicate whether a given item has the chosen tag. Actions that do not occur in the supervised portion of the dataset cannot be output by the model, but are retained for evaluation: For example, if during the bandit feedback phase, an example consists solely of tags that did not occur during the supervised phase, the algorithm will experience a reward of 0 for every feasible action on the example. For a typical seed, this results in roughly 890,000 feasible actions for the model. In the $(k = 5, r = 3)$ setup, we take the top-k actions as the greedy slate, and then independently decide whether to explore for each exploration slot (the bottom r slots). For exploration, we sample from the spanner set without replacement.

Regression functions and oracles. For bilinear models, regression functions take the form $f(x, a) = \langle \phi(a), Wx \rangle$, where W is a matrix of learned parameters. For deep models, regression functions pass the original context through 2 residual leaky ReLU layers before applying the bilinear layer, $f(x, a) = \langle \phi(a), W\bar{g}(x) \rangle$, where $\bar{g}$ is a learned two-layer neural network, and W is a matrix of learned parameters. For experiments with respect to oneshotwiki datasets, we add a learned bias term for regression functions (same for every action); for experiments with respect to the amazon-3m dataset, we additionally add an action-dependent bias term that is obtained from the supervised examples. The online regression oracle is implemented using PyTorch's Adam optimizer with log loss (recall that rewards are 0/1).

Hyperparameters. For each algorithm, we optimize its hyperparameters using random search (Bergstra and Bengio, 2012). Speccifically, hyperparameters are tuned by taking the best of 59 randomly selected configurations for a fixed seed (this seed is not used for evaluation). A seed determines both dataset shuffling, initialization of regressor parameters, and random choices made by any action sampling scheme.

Evaluation. We evaluate each algorithm on 32 seeds. All reported confidence intervals are 90% bootstrap CIs for the mean.

4.7.4.2 Practical Modification to Sampling Procedure in SpannerIGW

For experiments with SpannerIGW, we slightly modify the action sampling distribution so as to avoid computing the normalization constant λ. First, we modify the weighted embedding scheme given in Eq. (4.5) using the following expression:

$$\overline{\phi}(x_t, a) := \frac{\phi(x_t, a)}{\sqrt{1 + d + \frac{\gamma}{4d}\left(\widehat{f}_t(x_t, \widehat{a}_t) - \widehat{f}_t(x_t, a)\right)}}.$$

We obtain a $4d$-approximate optimal design for the reweighted embeddings by first computing a 2-approximate barycentric spanner $\mathcal{S}$, then taking $q_t^{\mathrm{opt}} := \mathrm{unif}(\mathcal{S})$. To proceed, let $\widehat{a}_t := \arg\max_{a \in \mathcal{A}} \widehat{f}(x_t, a)$ and $\overline{d} := |\mathcal{S} \cup \{\widehat{a}_t\}|$. We construct the sampling distribution $p_t \in \Delta(\mathcal{A})$ as follows:

- Set $p_t(a) := \frac{1}{\overline{d} + \frac{\gamma}{4d}\left(\widehat{f}_t(x_t, \widehat{a}_t) - \widehat{f}_t(x_t, a)\right)}$ for each $a \in \mathrm{supp}(\mathcal{S})$.
- Assign remaining probability mass to $\widehat{a}_t$.

With a small modification to the proof of Lemma 4.11, one can show that this construction certifies that $\mathrm{dec}_\gamma(\mathcal{F}) = O\left(\frac{d^2}{\gamma}\right)$. Thus, the regret bound in Theorem 4.12 holds up to a constant factor. Similarly, with a small modification to the proof of Theorem 4.13, we can also show that —with respect to this new embedding— Algorithm 11 has $O((\mathcal{T}_{\mathsf{Opt}} \cdot d^3 + d^4) \cdot \log^2\left(\frac{d+\gamma/d}{r}\right))$ runtime and $O(\mathcal{M}_{\mathsf{Opt}} + d^2 + d\log\left(\frac{d+\gamma/d}{r}\right))$ memory.

4.7.4.3 Timing Information

Table 4.4: Per-example inference timings for oneshotwiki-14031. CPU timings use batch size 1 on an Azure STANDARD_D4_V2 machine. GPU timings use batch size 1024 on an Azure STANDARD_NC6S_V2 (Nvidia P100-based) machine.

Algorithm	CPU	GPU
ε-Greedy	2 ms	10 µs
SpannerGreedy	2 ms	10 µs
SquareCB	2 ms	10 µs
SpannerIGW	25 ms	180 µs

Table 4.4 contains timing information the oneshotwiki-14031 dataset with a bilinear model. The CPU timings are most relevant for practical scenarios such as information retrieval and recommendation systems, while the GPU timings are relevant for scenarios where simulation is possible. Timings for SpannerGreedy do not include the one-time cost to compute the spanner set. Timings for all algorithms use precomputed context and action embeddings. For all but algorithms but SpannerIGW, timings reflect the major bottleneck of computing the argmax action, since all subsequent steps take $O(1)$ time with respect to $|\mathcal{A}|$. In particular, SquareCB is implemented using rejection sampling, which does not require explicit construction of the action distribution. For SpannerIGW, the additional overhead is due to the time required to construct an approximate optimal design for each example.

4.7.4.4 Additional Figures

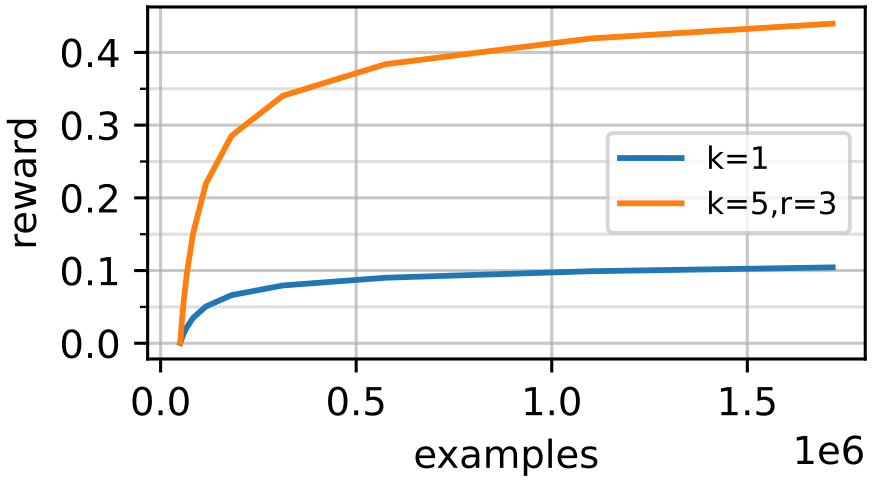


Figure 4.1: Performance of SpannerGreedy on amazon-3m.

In Fig. 4.1, we show the empirical performance of SpannerGreedy on amazon-3m. Confidence intervals are rendered, but are but too small to visualize. For $(k = 1)$, the final CI is $[0.1041, 0.1046]$, and for $(k = 5, r = 3)$, the final CI is $[0.438, 0.440]$.

5 CONTEXTUAL BANDITS WITH SMOOTH REGRET

Designing efficient general-purpose contextual bandit algorithms that work with large—or even continuous—action spaces would facilitate application to important scenarios such as information retrieval, recommendation systems, and continuous control. While obtaining standard regret guarantees can be hopeless, alternative regret notions have been proposed to tackle the large action setting. We propose a smooth regret notion for contextual bandits, which dominates previously proposed alternatives. We design a statistically and computationally efficient algorithm—for the proposed smooth regret—that works with general function approximation under standard supervised oracles. We also present an adaptive algorithm that automatically adapts to any smoothness level. Our algorithms can be used to recover the previous minimax/Pareto optimal guarantees under the standard regret, e.g., in bandit problems with multiple best arms and Lipschitz/Hölder bandits. We conduct large-scale empirical evaluations demonstrating the efficacy of our proposed algorithms.

5.1 Introduction

Contextual bandits concern the problem of sequential decision making with contextual information. Provably efficient contextual bandit algorithms have been proposed over the past decade (Langford and Zhang, 2007; Agarwal et al., 2014; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021; Foster and Krishnamurthy, 2021). However, these developments only work in setting with a small number of actions, and their theoretical guarantees become vacuous when working with a large action space (Agarwal et al., 2012). The hardness result can be intuitively understood through a "needle in the haystack" construction: When good actions are extremely rare, identifying any good action demands trying almost all alternatives. This prevents naive direct application of contextual bandit algorithms to large action problems, e.g., in information retrieval, recommendation systems, and

continuous control.

To bypass the hardness result, one approach is to assume structure on the model class. For example, in the standard linear contextual bandit (Auer, 2002; Chu et al., 2011; Abbasi-Yadkori et al., 2011), learning the d components of the reward vector—rather than examining every single action—effectively guides the learner to the optimal action. Additional structural assumptions have been studied in the literature, e.g., linearly structured actions and general function approximation (Foster et al., 2020a; Xu and Zeevi, 2020), Lipschitz/Hölder regression functions (Kleinberg, 2004; Hadiji, 2019), and convex functions (Lattimore, 2020). While these assumptions are fruitful theoretically, they might be violated in practice.

An alternative approach is to compete against a less demanding benchmark. Rather than competing against a policy that always plays the best action, one can compete against a policy that plays the best smoothed distribution over the actions: a smoothed distribution—by definition—cannot concentrate on the best actions when they are in fact rare. Thus, for the previously mentioned "needle in the haystack" construction, the benchmark is weak as well. This de-emphasizes such constructions and focuses algorithm design on scenarios where intuition suggests good solutions can be found without prohibitive statistical cost.

Contributions. We study large action space problems under an alternate notion of regret. Our first contribution is to propose a novel benchmark—the smooth regret—that formalizes the "no needle in the haystack" principle. We also show that our smooth regret dominates previously proposed regret notions along this line of work (Chaudhuri and Kalyanakrishnan, 2018; Krishnamurthy et al., 2020; Majzoubi et al., 2020), i.e., any regret guarantees with respect to the smooth regret automatically holds for these previously proposed regrets.

We design efficient algorithms that work with the smooth regret and general function classes. Our first proposed algorithm, SmoothIGW, works with any fixed smoothness level $h > 0$, and is efficient—both statistically and computationally—whenever the learner has access to standard oracles: (i) an online regression oracle for supervised learning, and (ii) a simple sampling oracle over the action space.

Statistically, SmoothIGW achieves $\sqrt{T/h}$-type regret for whatever action spaces; here $1/h$ should be viewed as the effective number of actions. Such guarantees can be verified to be minimax optimal when related back to the standard regret. Computationally, the guarantee is achieved with $O(1)$ operations with respect to oracles, which can be usually efficiently implemented in practice. Our second algorithm is a master algorithm which combines multiple SmoothIGW instances to compete against any unknown smoothness level. We show this master algorithm is Pareto optimal.

With our smooth regret and proposed algorithms, we exhibit guarantees under the standard regret in various scenarios, e.g., in problems with multiple best actions (Zhu and Nowak, 2020) and in problems when the expected payoff function satisfies structural assumptions such as Lipchitz/Hölder continuity (Kleinberg, 2004; Hadiji, 2019). Our algorithms are minimax/Pareto optimal when specialized to these settings.

5.1.1 Organization

We introduce our smooth regret in Section 5.2, together with statistical and computational oracles upon which our algorithms are built. In Section 5.3, we present our algorithm SmoothIGW, which illustrates the core ideas of learning with smooth regret at any fixed smoothness level. Built upon SmoothIGW, in Section 5.4, we present a CORRAL-type of algorithm that can automatically adapt to any unknown smoothness level. In Section 5.5, we connect our proposed smooth regret to the standard regret over various scenarios. We present empirical results in Section 5.6, and close with a discussion in Section 5.7. We defer most proofs to Section 5.8.

5.2 Problem Setting

We consider the following standard contextual bandit problems. At any time step $t \in [T]$, nature selects a context $x_t \in \mathcal{X}$ and a distribution over loss functions

$\ell_t : \mathcal{A} \to [0,1]$ mapping from the (compact) action set $\mathcal{A}$ to a loss value in $[0,1]$.[1] Conditioned on the context x_t, the loss function is stochastically generated, i.e., $\ell_t \sim \mathbb{P}_{\ell_t}(\cdot \mid x_t)$. The learner selects an action $a_t \in \mathcal{A}$ based on the revealed context x_t, and obtains (only) the loss $\ell_t(a_t)$ of the selected action. The learner has access to a set of measurable regression functions $\mathcal{F} \subseteq (\mathcal{X} \times \mathcal{A} \to [0,1])$ to predict the loss of any context-action pair. We make the following standard realizability assumption studied in the contextual bandit literature (Agarwal et al., 2012; Foster et al., 2018; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021).

Assumption 5.1 (Realizability)**.** *There exists a regression function* $f^\star \in \mathcal{F}$ *such that* $\mathbb{E}[\ell_t(a) \mid x_t] = f^\star(x_t, a)$ *for any* $a \in \mathcal{A}$ *and across all* $t \in [T]$.

The smooth regret. Let $(\mathcal{A}, \Omega)$ be a measurable space of the action set and μ be a base probability measure over the actions. Let $\mathcal{Q}_h$ denote the set of probability measures such that, for any measure $Q \in \mathcal{Q}_h$, the following holds true: (i) Q is absolutely continuous with respect to the base measure μ, i.e., $Q \ll \mu$; and (ii) The Radon-Nikodym derivative of Q with respect to μ is no larger than $\frac{1}{h}$, i.e., $\frac{dQ}{d\mu} \leqslant 1/h$. We call $\mathcal{Q}_h$ the set of smoothing kernels at smoothness level h, or simply put the set of h-smoothed kernels. For any context $x \in \mathcal{X}$, we denote by $\mathsf{Smooth}_h(x)$ the smallest loss incurred by any h-smoothed kernel, i.e.,

$$\mathsf{Smooth}_h(x) := \inf_{Q \in \mathcal{Q}_h} \mathbb{E}_{a \sim Q}[f^\star(x, a)].$$

Rather than competing with $\arg\min_{a \in \mathcal{A}} f^\star(x, a)$—an impossible job in many cases—we take $\mathsf{Smooth}_h(x)$ as the benchmark and define the *smooth regret* as follows:

$$\mathbf{Reg}_{\mathsf{CB},h}(T) := \mathbb{E}\left[\sum_{t=1}^{T} f^\star(x_t, a_t) - \mathsf{Smooth}_h(x_t)\right]. \tag{5.1}$$

[1] For the convenience of leveraging existing results, in this chapter, we consider loss functions instead of reward functions. Nevertheless, for any action $a \in \mathcal{A}$, its reward can be calculated as $r_t(a) = 1 - \ell_t(a) \in [0,1]$.

One important feature about the above definition is that the benchmark $\mathsf{Smooth}_h(x_t)$ automatically adapts to the context x_t: This gives the benchmark more power and makes it harder to compete against. In fact, our smooth regret dominates many existing regret measures with *easier* benchmarks. We provide some examples in the following.

- Chaudhuri and Kalyanakrishnan (2018) propose the quantile regret, which aims at competing with the lower h-quantile of the loss function, i.e., $\nu_h(x) := \inf\{\zeta : \mu(a \in \mathcal{A} : f^\star(x,a) \leqslant \zeta) \geqslant h\}$. Consider $\mathcal{S}_h := \{a \in \mathcal{A} : f^\star(x,a) \leqslant \nu_h(x)\}$ such that $\mu(\mathcal{S}_h) \geqslant h$. Let $\overline{Q}_h := \mu|_{\mathcal{S}_h}/\mu(\mathcal{S}_h)$ denote the (normalized) probability measure after restricting μ onto $\mathcal{S}_h$. Since $\overline{Q}_h \in \mathcal{Q}_h$, we clearly have $\mathsf{Smooth}_h(x) \leqslant \mathbb{E}_{a\sim\overline{Q}_h}[f^\star(x,a)] \leqslant \nu_h(x)$. Besides, the (original) quantile was only studied in the non-contextual case.

- Krishnamurthy et al. (2020) study a notion of regret that is smoothed in a different way: Their regret aims at competing with a known and *fixed* smoothing kernel (on top of a fixed policy set) with Radon-Nikodym derivative at most $1/h$. Our benchmark is clearly harder to compete against since we consider any smoothing kernel with Radon-Nikodym derivative at most $1/h$.

Besides being more competitive with respect to above benchmarks, smooth regret can also be naturally linked to the *standard* regret under various settings previously studied in the bandit literature, e.g., in the discrete case with multiple best arms (Zhu and Nowak, 2020) and in the continuous case with Lipschitz/Hölder continuous payoff functions (Kleinberg, 2004; Hadiji, 2019). We provide detailed discussion in Section 5.5.

5.2.1 Computational Oracles

The first step towards designing computationally efficient algorithms is to identify reasonable oracle models to access the sets of regression functions or actions. Otherwise, enumeration over regression functions or actions (both can be expo-

nentially large) immediately invalidate the computational efficiency. We consider two common oracle models: a regression oracle and a sampling oracle.

The regression oracles. A fruitful approach to designing efficient contextual bandit algorithms is through reduction to supervised regression with the class $\mathcal{F}$ (Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2021; Foster et al., 2020a, 2021a). Following Foster and Rakhlin (2020), we assume that we have access to an *online* regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$, which is an algorithm for sequential predication under square loss. More specifically, the oracle operates in the following protocol: At each round $t \in [T]$, the oracle makes a prediction $\widehat{f}_t$, then receives context-action-loss tuple $(x_t, a_t, \ell_t(a_t))$. The goal of the oracle is to accurately predict the loss as a function of the context and action, and we evaluate its performance via the square loss $(\widehat{f}_t(x_t, a_t) - \ell_t(a_t))^2$. We measure the oracle's cumulative performance through the square-loss regret to $\mathcal{F}$, which is formalized below.

Assumption 5.2. *The regression oracle* $\mathbf{Alg}_{\mathsf{Sq}}$ *guarantees that, with probability at least* $1-\delta$, *for any* (*potentially adaptively chosen*) *sequence* $\{(x_t, a_t, \ell_t(a_t))\}_{t=1}^T$,

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(\widehat{f}_t(x_t, a_t) - \ell_t(a_t)\right)^2 - \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}(f(x_t, a_t) - \ell_t(a_t))^2\right] \leqslant \mathbf{Reg}_{\mathsf{Sq}}(T, \delta),$$

for some (*non-data-dependent*) *function* $\mathbf{Reg}_{\mathsf{Sq}}(T, \delta)$.

Sometimes it's useful to consider a *weighted* regression oracle, where the square errors are weighted differently. It is shown in Foster et al. (2020a) (Theorem 5 therein) that any regression oracle satisfies Assumption 5.2 can be used to generate a weighted regression oracle that satisfies the following assumption.

Assumption 5.3. *The regression oracle* $\mathbf{Alg}_{\mathsf{Sq}}$ *guarantees that, with probability at least* $1-\delta$*, for any (potentially adaptively chosen) sequence* $\{(w_t, x_t, a_t, \ell_t(a_t))\}_{t=1}^{T}$*,*

$$\mathbb{E}\left[\sum_{t=1}^{T} w_t\left(\widehat{f}_t(x_t, a_t) - \ell_t(a_t)\right)^2 - \inf_{f\in\mathcal{F}}\sum_{t=1}^{T} w_t(f(x_t, a_t) - \ell_t(a_t))^2\right] \leqslant \mathbb{E}\left[\max_{t\in[T]} w_t\right]\mathbf{Reg}_{\mathsf{Sq}}(T, \delta),$$

for some (non-data-dependent) function $\mathbf{Reg}_{\mathsf{Sq}}(T, \delta)$*.*

For either regression oracle, we let $\mathcal{T}_{\mathsf{Sq}}$ denote an upper bound on the time to (i) query the oracle's estimator $\widehat{f}_t$ with context-action pair (x_t, a) and receive its predicated value $\widehat{f}_t(x_t, a) \in [0, 1]$; (ii) query the oracle's estimator $\widehat{f}_t$ with context x_t and receive its argmin action $\widehat{a}_t = \arg\min_{a\in\mathcal{A}} \widehat{f}_t(x_t, a)$; and (iii) update the oracle with example $(x_t, a_t, r_t(a_t))$. We let $\mathcal{M}_{\mathsf{Sq}}$ denote the maximum memory used by the oracle throughout its execution.

Online regression is a well-studied problem, with known algorithms for many model classes (Foster and Rakhlin, 2020; Foster et al., 2020a): including linear models (Hazan et al., 2007), generalized linear models (Kakade et al., 2011), non-parametric models (Gaillard and Gerchinovitz, 2015), and beyond. Using Vovk's aggregation algorithm (Vovk, 1998), one can show that $\mathbf{Reg}_{\mathsf{Sq}}(T, \delta) = O(\log(|\mathcal{F}|/\delta))$ for any finite set of regression functions $\mathcal{F}$, which is the canonical setting studied in contextual bandits (Langford and Zhang, 2007; Agarwal et al., 2012). In the following of this chapter, we use abbreviation $\mathbf{Reg}_{\mathsf{Sq}}(T) := \mathbf{Reg}_{\mathsf{Sq}}(T, T^{-1})$, and will keep the $\mathbf{Reg}_{\mathsf{Sq}}(T)$ term in our regret bounds to accommodate for general set of regression functions.

The sampling oracles. In order to design algorithms that work with large/continuous action spaces, we assume access to a sampling oracle $\mathbf{Alg}_{\mathsf{Sample}}$ to get access to the action space. In particular, the oracle $\mathbf{Alg}_{\mathsf{Sample}}$ returns an action $a \sim \mu$ randomly drawn according to the base probability measure μ over the action space $\mathcal{A}$. We let

$\mathcal{T}_{\mathsf{Sample}}$ denote a bound on the runtime of single query to the oracle; and let $\mathcal{M}_{\mathsf{Sample}}$ denote the maximum memory used by the oracle.

Representing the actions. We use $b_{\mathcal{A}}$ to denote the number of bits required to represent any action $a \in \mathcal{A}$, which scales with $O(\log|\mathcal{A}|)$ with a finite set of actions and $\widetilde{O}(d)$ for actions represented as vectors in $\mathbb{R}^d$. Tighter bounds are possible with additional structual assumptions. Since representing actions is a minimal assumption, we hide the dependence on $b_{\mathcal{A}}$ in big-O notation for our runtime and memory analysis.

5.3 Efficient Algorithm with Smooth Regret

We design an oracle-efficient (SmoothIGW, Algorithm 14) algorithm that achieves a $\sqrt{T}$-type regret under the smooth regret defined in Eq. (5.1). We focus on the case when the smoothness level $h > 0$ is known in this section, and leave the design of adaptive algorithms in Section 5.4.

Algorithm 14 contains the pseudo code of our proposed SmoothIGW algorithm, which deploys a smoothed sampling distribution to balance exploration and exploitation. At each round $t \in [T]$, the learner observes the context x_t from the environment and obtains the estimator $\widehat{f}_t$ from the regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$. It then constructs a sampling distribution P_t by mixing a smoothed distribution constructed using the *inverse gap weighting* (IGW) technique (Abe and Long, 1999; Foster and Rakhlin, 2020) and a delta mass at the greedy action $\widehat{a}_t := \arg\min_{a\in\mathcal{A}} \widehat{f}_t(x_t, a)$. The algorithm samples an action $a_t \sim P_t$ and then update the regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$. The key innovation of the algorithm lies in the construction of the smoothed IGW distribution, which we explain in detail next.

Smoothed variant of IGW. The IGW technique was previously used in the finite-action contextual bandit setting (Abe and Long, 1999; Foster and Rakhlin, 2020), which assigns a probability mass to every action $a \in \mathcal{A}$ inversely proportional to the estimated loss gap $(\widehat{f}(x, a) - \widehat{f}(x, \widehat{a}))$. To extend this strategy to continuous action

Algorithm 14 SmoothIGW

Input: Exploration parameter $\gamma > 0$, online regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Observe context x_t.
3: Receive $\widehat{f}_t$ from regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
4: Get $\widehat{a}_t \leftarrow \arg\min_{a \in \mathcal{A}} \widehat{f}_t(x_t, a)$.
5: Define

$$P_t := M_t + (1 - M_t(\mathcal{A})) \cdot \mathbb{I}_{\widehat{a}_t}, \tag{5.2}$$

where M_t is the measure defined in Eq. (5.4)
6: Sample $a_t \sim P_t$ and observe loss $\ell_t(a_t)$. `// This can be done efficiently via Algorithm 15.`
7: Update $\mathbf{Alg}_{\mathsf{Sq}}$ with $(x_t, a_t, \ell_t(a_t))$

spaces we leverage Radon-Nikodym derivatives. Fix any constant $\gamma > 0$, we define a IGW-type function as

$$m_t(a) := \frac{1}{1 + h\gamma(\widehat{f}_t(x_t, a) - \widehat{f}_t(x_t, \widehat{a}_t))}. \tag{5.3}$$

For any $\omega \in \Omega$, we then define a new measure

$$M_t(\omega) := \int_{a \in \omega} m_t(a)\, d\mu(a) \tag{5.4}$$

of the measurable action space $(\mathcal{A}, \Omega)$, where $m(a) = \frac{dM}{d\mu}(a)$ serves as the Radon-Nikodym derivative between the new measure M and the base measure μ. Since $m_t(a) \leqslant 1$ by construction, we have $M_t(\mathcal{A}) \leqslant 1$, i.e., M_t is a sub-probability measure. SmoothIGW plays a probability measure $P_t \in \Delta(\mathcal{A})$ by mixing the sub-probability measure M_t with a delta mass at the greedy action $\widehat{a}_t$, as in Eq. (5.2).

Efficient sampling. We now discuss how to sample from the distribution of Eq. (5.2) using a single call to the sampling oracle, via rejection sampling. We first randomly sample an action $a \sim \mu$ from the sampling oracle $\mathbf{Alg}_{\mathsf{Sample}}$ and

Algorithm 15 Rejection Sampling for IGW

Input: Sampling oracle $\mathbf{Alg}_{\mathsf{Sample}}$, greedy action $\widehat{a}_t$, Radon-Nikodym derivative $m_t(a)$.

Draw $a \sim \mu$ from sampling oracle $\mathbf{Alg}_{\mathsf{Sample}}$.
Sample Z from a Bernoulli random distribution with mean $m_t(a)$.
if $Z = 1$ **then**
    Take action a.
else
    Take action $\widehat{a}_t$.

with respect to the base measure μ. We then compute $m_t(a)$ in Eq. (5.3) with two evaluation calls to $\widehat{f}_t$, one at $\widehat{f}_t(x_t, a)$ and the other at $\widehat{f}_t(x_t, \widehat{a}_t)$. Finally, we sample a random variable Z from a Bernoulli distribution with expectation $m_t(a)$ and play either action $\widehat{a}_t$ or action a depending upon the realization of Z. One can show that the sampling distribution described above coincides with the distribution defined in Eq. (5.2) (Proposition 5.4).[2] We present the pseudo code for rejection sampling in Algorithm 15.

Proposition 5.4. *The sampling distribution generated from Algorithm 15 coincides with the sampling distribution defined in Eq. (5.2).*

Proof of Proposition 5.4. Let $\overline{P}_t$ denote the sampling distribution achieved by Algorithm 15. For any $\omega \in \Omega$, if $\widehat{a}_t \notin \omega$, we have

$$\overline{P}_t(\omega) = \int_{a \in \omega} m_t(a)\, d\mu(a) = M_t(\omega)$$

Now suppose that $\widehat{a}_t \in \omega$: Then the rejection probability, which equals $\mathbb{E}_{a\sim\mu}[1 - m_t(a)] = 1 - M_t(\mathcal{A})$, will be added to the above expression. □

We now state the regret bound for SmoothIGW in the following.

[2] The same idea can be immediately applied to the case of sampling from the IGW distribution with finite number of actions (Foster and Rakhlin, 2020).

Theorem 5.5. *Fix any smoothness level* $h \in (0, 1]$. *With an appropriate choice for* $\gamma > 0$, *Algorithm 14 ensures that*

$$\mathbf{Reg}_{\mathsf{CB},h}(T) \leqslant \sqrt{4T\,\mathbf{Reg}_{\mathsf{Sq}}(T)/h},$$

with per-round runtime $O(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Sample}})$ *and maximum memory* $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Sample}})$.

Key features of Algorithm 14. Algorithm 14 achieves $\widetilde{O}(\sqrt{T/h})$ regret, which has no dependence on the number of actions.[3] This suggests the Algorithm 14 can be used in large action spaces scenarios and only suffer regret scales with $1/h$: the effective number of actions considered for smooth regret. We next highlight the statistical and computational efficiencies of Algorithm 14.

- *Statistical optimality.* It's not hard to prove a $\widetilde{\Omega}(\sqrt{T/h})$ lower bound for the smooth regret by relating it to the standard regret under a contextual bandit problem with finite actions: (i) the smooth regret and the standard regret coincides when $h = 1/|\mathcal{A}|$; and (ii) the standard regret admits lower bound $\widetilde{\Omega}(\sqrt{|\mathcal{A}|T})$ (Agarwal et al., 2012). In Section 5.5, we further relate our smooth regret guarantee to standard regret guarantee under other scenarios and recover the minimax bounds.

- *Computational efficiency.* Algorithm 14 is oracle-efficient and enjoys per-round runtime and maximum memory that scales linearly with oracle costs. To our knowledge, this leads to the first computationally efficient general-purpose algorithm that achieves a $\sqrt{T}$-type guarantee under smooth regret. The previously known efficient algorithm applies an ε-Greedy-type of strategy and thus only achieves a $T^{2/3}$-type regret (Majzoubi et al. (2020), and with respect to a weaker version of the smooth regret).

Proof sketch for Theorem 5.5. To analyze Algorithm 14, we follow a recipe introduced by Foster and Rakhlin (2020); Foster et al. (2020a, 2021b) based on the

[3] We focus on the canonical case studied in contextual bandits with a finite $\mathcal{F}$, and view $\mathbf{Reg}_{\mathsf{Sq}}(T) = O(\log|\mathcal{F}|)$.

Decision-Estimation Coefficient (DEC, adjusted to our setting), defined as $\mathsf{dec}_\gamma(\mathcal{F}) := \sup_{\widehat{f},x} \mathsf{dec}_\gamma(\mathcal{F}; \widehat{f}, x)$, where

$$\mathsf{dec}_\gamma(\mathcal{F}; \widehat{f}, x) := \inf_{P \in \Delta(\mathcal{A})} \sup_{f^\star \in \mathcal{F}} \mathbb{E}_{a \sim P}\left[f^\star(x, a^\star) - \mathsf{Smooth}_h(x) - \frac{\gamma}{4} \cdot \left(\widehat{f}(x, a) - f^\star(x, a)\right)^2\right]. \tag{5.5}$$

Foster and Rakhlin (2020); Foster et al. (2020a, 2021b) consider a meta-algorithm which, at each round t, (i) computes $\widehat{f}_t$ by appealing to a regression oracle, (ii) computes a distribution $P_t \in \Delta(\mathcal{A})$ that solves the minimax problem in Eq. (5.5) with x_t and $\widehat{f}_t$ plugged in, and (iii) chooses the action a_t by sampling from this distribution. One can show that for any $\gamma > 0$, this strategy enjoys the following regret bound:

$$\mathbf{Reg}_{\mathsf{CB},h}(T) \lesssim T \cdot \mathsf{dec}_\gamma(\mathcal{F}) + \gamma \cdot \mathbf{Reg}_{\mathsf{Sq}}(T), \tag{5.6}$$

More generally, if one computes a distribution that does not solve Eq. (5.5) exactly, but instead certifies an upper bound on the DEC of the form $\mathsf{dec}_\gamma(\mathcal{F}) \leqslant \overline{\mathsf{dec}}_\gamma(\mathcal{F})$, the same result holds with $\mathsf{dec}_\gamma(\mathcal{F})$ replaced by $\overline{\mathsf{dec}}_\gamma(\mathcal{F})$. Algorithm 14 is a special case of this meta-algorithm, so to bound the regret it suffices to show that the exploration strategy in the algorithm certifies a bound on the DEC.

By applying principles of convex conjugate, we show that the IGW-type distribution of Eq. (5.2) certifies $\mathsf{dec}_\gamma(\mathcal{F}) \leqslant \frac{2}{h\gamma}$ for any set of regression functions $\mathcal{F}$ (Lemma 5.12, deferred to Section 5.8.1.1). With this bound on DEC, We can then bound the first term in Eq. (5.6) by $O(\frac{T}{h\gamma})$ and optimally tune γ in Eq. (5.6) to obtain the desired regret guarantee.

Deriving the bound on the DEC is one of our key technical contributions, where we simultaneous eliminate the dependence on both the function class and (cardinality of) the action set. Previous bounds on the DEC assume either a restricted function class $\mathcal{F}$ or a finite action set.

5.4 Adapting to Unknown Smoothness Parameters

Our results in Section 5.3 shows that, with a known h, one can achieve smooth regret proportional to $\sqrt{T/h}$ against the optimal smoothing kernel in $\mathcal{Q}_h$. The total loss achieved by the learner is the smooth regret plus the total loss suffered by playing the optimal smoothing kernel. One can notice that these two terms go into different directions: When h gets smaller, the loss suffered by the optimal smoothing kernel gets smaller, yet the regret term gets larger. It is apriori unclear how to balance these terms, and therefore desirable to design algorithms that can automatically adapt to an unknown $h \in (0, 1]$. Note it is sufficient to adapt to unknown $h \in [1/T, 1]$, as the regret bound is vacuous for $h < 1/T$. We provide such an algorithm in this section.

The CORRAL master algorithm. Our algorithm follows the standard master-base algorithm structure: We run multiple base algorithms with different configurations in parallel, and then use a master algorithm to conduct model selection on top of base algorithms. The goal of the master algorithm is to balance the regret among base algorithms and eventually achieve a performance that is "close" to the best base algorithm (whose identity is unknown). We use the classical CORRAL algorithm (Agarwal et al., 2017) as the master algorithm and initiate a collection of $B = \lceil \log T \rceil$ (modified) Algorithm 14 as base algorithms. More specifically, for $b = 1, 2, \ldots, B$, each base algorithm is initialized with smoothness level $h_b = 2^{-b}$. For any $h^\star \in [1/T, 1]$, one can notice that there exists a base algorithm $i^\star$ that suits well to this (unknown) $h^\star$ in the sense that $h_{b^\star} \leqslant h^\star \leqslant 2h_{b^\star}$. The goal of the master algorithm is thus to adapt to the base algorithm indexed by $b^\star$.

We provide a brief description of the CORRAL master algorithm, and direct the reader to Agarwal et al. (2017) for more details. The master algorithm maintains a distribution $q_t \in \Delta([B])$ over base algorithms. At each round, the master algorithm sample a base algorithm $I_t \sim q_t$ and passes the context x_t, the sampling probability q_{t,I_t} and parameter $\rho_{t,I_t} := 1/\min_{i \leqslant t} q_{t,I_t}$ into the base algorithm I_t. The base algorithm I_t then performs its learning process: it samples an arm a_t, observes its

loss $\ell_t(a_{t,I_t})$, and then updates its internal state. The master algorithm is updated with respect to the importance-weighted loss $\frac{\ell_t(a_{t,I_t})}{q_{t,I_t}}$ and parameter ρ_{t,I_t}. In order to obtain theoretical guarantees, the base algorithms are required to be stable, which is defined as follows.

Definition 5.6. *Suppose the base algorithm indexed by* b *satisfies—when implemented alone—regret guarantee* $\mathbf{Reg}_{\mathsf{CB},h_b}(T) \leqslant R_b(T)$ *for some non-decreasing* $R_b(T) : \mathbb{N}_+ \to \mathbb{R}_+$. *Let* $\mathbf{Reg}_{\mathsf{Imp},h}$ *denote the* importance-weighted *regret for base algorithm* b, *i.e.,*

$$\mathbf{Reg}_{\mathsf{Imp},h_b}(T) := \mathbb{E}\left[\sum_{t=1}^{T} \frac{\mathbb{1}(I_t = b)}{q_{t,b}}(f^\star(x_t, a_t) - \mathsf{Smooth}_{h_b}(x_t))\right].$$

The base algorithm b *is called* $(\alpha, R_b(T))$ *stable if* $\mathbf{Reg}_{\mathsf{Imp},h_b}(T) \leqslant \mathbb{E}\left[\rho_{T,b}^{\alpha}\right] R_b(T)$.

A stable base algorithm. Our treatment is inspired by Foster et al. (2020a). Let $(\tau_1, \tau_2, \ldots) \subseteq [T]$ denote the time steps when the base algorithm b is invoked, i.e., when $I_t = b$. When invoked, the base algorithm receives $(x_t, q_{t,b}, \rho_{t,b})$ from the master algorithm. The base algorithm then sample from a distribution similar to Eq. (5.2) but with a customized learning rate $\gamma_{t,b} := \sqrt{8T/(h_b \cdot \rho_{t,b} \cdot \mathbf{Reg}_{\mathsf{Sq}}(T))}$. After observing the loss $\ell_t(a_{t,b})$, the base algorithm then updates the weighted regression oracle satisfying Assumption 5.3. Our modified algorithm is summarized in Algorithm 16.

Algorithm 16 Stable Base Algorithm (Index b)

Input: Weighted online regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.

Initialize weighted regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
for $t \in (\tau_1, \tau_2, \ldots)$ **do**
  Receive context x_t, probability $q_{t,b}$ and parameter $\rho_{t,b}$ from the master algorithm.
  Receive $\widehat{f}_{t,b}$ from the *weighted* online regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$.
  Get $\widehat{a}_{t,b} \leftarrow \arg\min_{a \in \mathcal{A}} \widehat{f}_{t,b}(x_t, a)$.
  Define $\gamma_{t,b} := \sqrt{8T/(h_b \cdot \rho_{t,b} \cdot \mathbf{Reg}_{\mathsf{Sq}}(T))}$ and $w_{t,b} := \mathbb{1}(I_t = b) \cdot \gamma_{t,b}/q_{t,b}$.
  Define $P_{t,b} := M_{t,b} + (1 - M_{t,b}(\mathcal{A})) \cdot \mathbb{I}_{\widehat{a}_{t,b}}$ according to Eq. (5.2) but with $\gamma_{t,b}$ defined above.
  Sample $a_{t,b} \sim P_{t,b}$ and observe loss $\ell_t(a_{t,b})$. `// This can be done efficiently via Algorithm 15.`
  Update the weighted regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$ with $(w_{t,b}, x_t, a_t, \ell_t(a_{t,b}))$

Proposition 5.7. *For any $b \in [B]$, Algorithm 16 is $\left(\frac{1}{2}, \sqrt{4T\,\mathbf{Reg}_{\mathsf{Sq}}(T)/h_b}\right)$-stable, with per-round runtime $O(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Sample}})$ and maximum memory $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Sample}})$.*

We now provide our model selection guarantees that adapt to unknown smoothness parameter $h \in (0, 1]$. The result directly follows from combining the guarantee of CORRAL (Agarwal et al., 2017) and our stable base algorithms.

Theorem 5.8. *Fix learning rate $\eta \in (0, 1]$, the CORRAL algorithm with Algorithm 16 as base algorithms guarantees that*

$$\mathbf{Reg}_{\mathsf{CB},h}(T) = \widetilde{O}\left(\frac{1}{\eta} + \frac{\eta\, T\, \mathbf{Reg}_{\mathsf{Sq}}(T)}{h}\right), \forall h \in (0, 1].$$

The CORRAL master algorithm has per-round runtime $\widetilde{O}(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Sample}})$ and maximum memory $\widetilde{O}(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Sample}})$.

Remark 5.9. *We keep the current form of Theorem 5.8 to better generalize to other settings, as explained in Section 5.5. With a slightly different analysis, we can recover the $\widetilde{O}(T^{\frac{1}{1+\beta}} h^{-\beta} (\log|\mathcal{F}|)^{\frac{\beta}{1+\beta}})$ guarantee for any $\beta \in [0, 1]$, which is known to be Pareto optimal (Krishnamurthy et al., 2020). We provide the proofs for this result in Section 5.8.2.2.*

5.5 Extensions to Standard Regret

We extend our results to various settings under the standard regret guarantee, including the discrete case with multiple best arms, and the continuous case under Lipschitz/Hölder continuity. Our results not only recover previously known minimax/Pareto optimal guarantees, but also generalize existing results in various ways.

Although our guarantees are stated in terms of the smooth regret, they are naturally linked to the standard regret among various settings studied in this section. We thus primarily focus on the standard regret in this section. Let $a_t^\star := \arg\min_{a \in \mathcal{A}} f^\star(x_t, a)$ denote the best action under context x_t. The *standard* (expected) regret is defined as

$$\mathbf{Reg}_{\mathsf{CB}}(T) := \mathbb{E}\left[\sum_{t=1}^{T} f^\star(x_t, a_t) - f^\star(x_t, a_t^\star)\right].$$

We focus on the canonical case with a finite set of regression functions $\mathcal{F}$ and consider $\mathbf{Reg}_{\mathsf{Sq}}(\mathcal{F}) = O(\log(|\mathcal{F}|T))$ (Vovk, 1998).

5.5.1 Discrete Case: Bandits with Multiple Best Arms

Zhu and Nowak (2020) study a non-contextual bandit problem with a large (discrete) action set $\mathcal{A}$ which might contain multiple best arms. More specifically, suppose there exists a subset of optimal arms $\mathcal{A}^\star \subseteq \mathcal{A}$ with cardinalities $|\mathcal{A}^\star| = K^\star$ and $|\mathcal{A}| = K$, the goal is to adapt to the effective number of arms $\frac{K}{K^\star}$ and minimize the standard regret. Note that one could have $\frac{K}{K^\star} \ll K$ when $K^\star$ is large.

Existing Results. Suppose $\frac{K}{K^\star} = \Theta(T^\alpha)$ for some $\alpha \in [0, 1]$. Zhu and Nowak (2020) shows that: (i) when α is known, the minimax regret is $\widetilde{\Theta}(T^{(1+\alpha)/2})$; and (ii) when α is unknown, the Pareto optimal regret can be described by $\widetilde{O}(\max\{T^\beta, T^{1+\alpha-\beta}\})$ for any $\beta \in [0, 1)$.

Our Generalizations. We extend the problem to the contextual setting: We use $\mathcal{A}^\star_{x_t} \subseteq \mathcal{A}$ to denote the *subset* of optimal arms with respect to context x_t, and

analogously assume that $\inf_{x \in \mathcal{X}} |\mathcal{A}_x^\star| = K^\star$ and $\frac{K}{K^\star} = T^\alpha$.

Since $\frac{K^\star}{K}$ represents the proportion of actions that are optimal, by setting $h = \frac{K^\star}{K} = T^{-\alpha}$ (and under uniform measure), we can then relate the standard regret to the smooth regret, i.e., $\mathbf{Reg}_{\mathsf{CB}}(T) = \mathbf{Reg}_{\mathsf{CB},h}(T)$. In the case when α is known, Theorem 5.5 implies that $\mathbf{Reg}_{\mathsf{CB}}(T) = O\big(T^{(1+\alpha)/2} \log^{1/2}(|\mathcal{F}|T)\big)$. In the case with unknown α, by setting $\eta = T^{-\beta}$ in Theorem 5.8, we have

$$\mathbf{Reg}_{\mathsf{CB}}(T) = O\big(\max(T^\beta, T^{1+\alpha-\beta} \log(|\mathcal{F}|T))\big).$$

These results generalize the known minimax/Pareto optimal results in Zhu and Nowak (2020) to the contextual bandit case, up to logarithmic factors.

5.5.2 Continuous Case: Lipschitz/Hölder Bandits

Kleinberg (2004); Hadiji (2019) study non-contextual bandit problems with (non-contextual) mean payoff functions $f^\star(a)$ satisfying Hölder continuity. More specifically, let $\mathcal{A} = [0, 1]$ (with uniform measure) and $L, \alpha > 0$ be some Hölder smoothness parameters, the assumption is that

$$|f^\star(a) - f^\star(a')| \leqslant L\,|a - a'|^\alpha,$$

for any $a, a' \in \mathcal{A}$. The goal is to adapt to provide standard regret guarantee that adapts to the smoothness parameters L and α.

Existing Results. In the case when L, α are known, Kleinberg (2004) shows that the minimax regret scales as $\Theta(L^{1/(2\alpha+1)} T^{(\alpha+1)/(2\alpha+1)})$; in the case with unknown L, α, Hadiji (2019) shows that the Pareto optimal regret can be described by $\widetilde{O}\big(\max\{T^\beta, L^{1/(1+\alpha)} T^{1-\frac{\alpha}{1+\alpha}\beta}\}\big)$ for any $\beta \in [\frac{1}{2}, 1]$.

Our Generalizations. We extend the setting to the contextual bandit case and

make the following analogous Hölder continuity assumption,[4] i.e.,

$$|f^\star(x, a) - f^\star(x, a')| \leqslant L\,|a - a'|^\alpha, \quad \forall x \in \mathcal{X}.$$

We first divide the action set $\mathcal{A} = [0, 1]$ into $B = \lceil 1/h \rceil$ consecutive intervals $\{I_b\}_{b=1}^B$ such that $I_b = [(b-1)h, bh]$. Let b_t denote the index of the interval where the best action $a_t^\star := \arg\min_{a \in \mathcal{A}} f^\star(x_t, a)$ lies into, i.e., $a_t^\star \in I_{b_t}$. Our smooth regret (at level h) provides guarantees with respect to the smoothing kernel $\mathrm{unif}(I_{b_t})$. Since we have $\mathbb{E}_{a \sim \mathrm{unif}(I_{b_t})}[f^\star(x_t, a)] \leqslant f^\star(x_t, a_t^\star) + Lh^\alpha$ under Hölder continuity, the following guarantee holds under the standard regret

$$\mathbf{Reg}_{\mathrm{CB}}(T) \leqslant \mathbf{Reg}_{\mathrm{CB},h}(T) + Lh^\alpha T. \tag{5.7}$$

When L, α are known, setting $h = \Theta\big(L^{-2/(2\alpha+1)} T^{-1/(2\alpha+1)} \log^{1/(2\alpha+1)}(|\mathcal{F}|T)\big)$ in Theorem 5.5 (together with Eq. (5.7)) leads to regret guarantee $O\big(L^{1/(2\alpha+1)} T^{(\alpha+1)/(2\alpha+1)} \log^{(\alpha/(2\alpha+1)}(|\mathcal{F}|T)\big)$, which is nearly minimax optimal (Kleinberg, 2004). In the case when L, α are unknown, setting $\eta = T^{-\beta}$ in Theorem 5.8 (together with Eq. (5.7)) leads to

$$\mathbf{Reg}_{\mathrm{CB}}(T) = O\Big(\max\Big\{T^\beta, L^{1/(1+2\alpha)} T^{1-\frac{\alpha}{1+\alpha}\beta} \log^{\alpha/(1+\alpha)}(|\mathcal{F}|T)\Big\}\Big),$$

which matches the Pareto frontier obtained in Hadiji (2019) up to logarithmic factors.

5.6 Experiments

In this section we compare our technique empirically with prior art from the bandit and contextual bandit literature. Code to reproduce these experiments is available at `https://github.com/pmineiro/smoothcb`.

[4] The special case with Lipschitz continuity ($\alpha = 1$) has been previously studied in the contextual setting, e.g., see Krishnamurthy et al. (2020).

5.6.1 Comparison with Bandit Prior Art

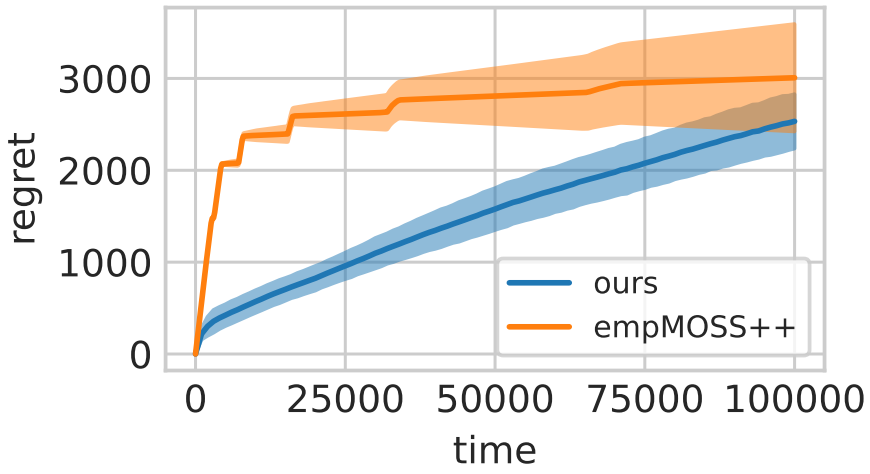


Figure 5.1: Comparison of regret on a bandit dataset with a discrete action space.

We replicate the real-world dataset experiment from Zhu and Nowak (2020). The dataset consists of 10025 captions from the *New Yorker Magazine* Cartoon Caption Contest and associated average ratings, normalized to $[0, 1]$. The caption text is discarded resulting in a non-contextual bandit problem with 10025 arms. When an arm is chosen, the algorithm experiences a Bernoulli loss realization whose mean is one minus the average rating for that arm. The goal is to experience minimum regret over the planning horizon $T = 10^5$. There are 54 arms in the dataset that have the minimal mean loss of 0.

For our algorithm, we used the uniform distribution over $[1, 2, \ldots, |\mathcal{A}|]$ as a reference measure, for which $O(1)$ sampling is available. We instantiated a tabular regression function, i.e., for each arm we maintained the empirical loss frequency observed for that arm. We use CORRAL with learning rate $\eta = 1$ and instantiated 8 subalgorithms with γ_h geometrically evenly spaced between 10^3 and 10^6. These were our initial hyperparameter choices, but they worked well enough that no tuning was required.

In Fig. 5.1, we compare our technique with `empMOSS++`, the best performing technique from Zhu and Nowak (2020). We plot the regret for both algorithms (smaller is better). Following the display convention of Zhu and Nowak (2020), shaded areas in the plot represent 0.5 standard deviation (i.e., it captures around 38% confidence region). Our technique is statistically equivalent.

5.6.2 Comparison with Contextual Bandit Prior Art

We replicate the online setting from Majzoubi et al. (2020), where 5 large-scale OpenML regression datasets are converted into continuous action problems on $[0, 1]$ by shifting and scaling the target values into this range. The context x is a mix of numerical and categorical variables depending upon the particular OpenML dataset. For any example, when the algorithm plays action a and the true target is y, the algorithm experiences loss $|y - a|$ as bandit feedback.

We use Lebesgue measure on $[0, 1]$ as our reference measure, for which $O(1)$ sampling is available. To maintain $O(1)$ computation, we consider regression functions with (learned) parameters θ via $f(x, a; \theta) := g\left(\widehat{a}\left(x; \theta\right) - a; \theta\right)$ where, for any θ, $z = 0$ is a global minimizer of $g(z; \theta)$. Subject to this constraint, we are free to choose $g(\cdot; \theta)$ and $\widehat{a}(\cdot; \theta)$ and yet are ensured that we can directly compute the minimizer of our loss predictor via $\widehat{a}(x; \theta)$. For our experiments we use a logistic loss predictor and a linear argmin predictor with logistic link: Let $\theta := (v; w; \xi)$, we choose

$$g(z; \theta) := \sigma\left(|w||z| + \xi\right), \quad \text{and} \quad \widehat{a}(x; \theta) := \sigma\left(v^\top x\right),$$

where $\sigma(\cdot)$ is the sigmoid function.

In Table 5.1, we compare our technique with CATS from Majzoubi et al. (2020). Following their protocol, we tune hyperparameters for each dataset to be optimal in-hindsight, and then report 95% bootstrap confidence intervals based upon the progressive loss of a single run. Our algorithm outperforms CATS.

To further exhibit the generality of our technique, we also include results for a nonlinear argmin predictor in Table 5.1 (last column), which uses a Laplace kernel

Table 5.1: Average progressive loss on contextual bandits datasets with continuous action spaces, scaled by 1000.

	CATS	Ours (Linear)	Ours (RFF)
Cpu	[55, 57]	[40.6, 40.7]	[**38.6**, **38.7**]
Fri	[183, 187]	[161, 163]	[**156**, **157**]
Price	[108, 110]	[70.2, 70.5]	[**66.1**, **66.3**]
Wis	[172, 174]	[138, 139]	[**136.2**, **136.6**]
Zur	[24, 26]	[24.3, 24.4]	[25.4, 25.5]

regressor implemented via random Fourier features (Rahimi et al., 2007) to predict the argmin. This approach achieves even better empirical performance.

5.7 Discussion

This work presents simple and practical algorithms for contextual bandits with large—or even continuous—action spaces, continuing a line of research which assumes actions that achieve low loss are not rare. While our approach can be used to recover minimax/Pareto optimal guarantees under certain structural assumptions (e.g., with Hölder/Lipschitz continuity), it doesn't cover all cases. For instance, on a large but finite action set with a linear reward function, the optimal smoothing kernel can be made to perform arbitrarily worse than the optimal action (e.g., by having one optimal action lying in an orthogonal space of all other actions); in this construction, algorithms provided in this chapter would perform poorly relative to specialized linear contextual bandit algorithms.

In future work we will focus on offline evaluation. Our technique already generates data that is suitable for subsequent offline evaluation of policies absolutely continuous with the reference measure, but only when the submeasure sample is accepted (line 4 of Algorithm 15), i.e., only $M(\mathcal{A})$ fraction of the data is suitable for reuse. We plan to refine our sampling distribution so that the fraction of re-usable data can be increased, but presumably at the cost of additional computation.

We manage to achieve a $\sqrt{T}$-regret guarantee with respect to smooth regret, which dominates previously studied regret notions that competing against easier benchmarks. A natural question to ask is, what is the strongest benchmark such that it is possible to still achieve a $\sqrt{T}$-type guarantee for problems with arbitrarily large action spaces? Speculating, there might exist a regret notion which dominates smooth regret yet still admits a $\sqrt{T}$ guarantee.

5.8 Proofs and Supporting Results

5.8.1 Proofs and Supporting Results for Section 5.3

This section is organized as follows. We provide supporting results in Section 5.8.1.1, then give the proof of Theorem 5.5 in Section 5.8.1.2.

5.8.1.1 Supporting Results

Preliminaries.

We first introduce the concept of convex conjugate. For any function $\phi : \mathbb{R} \to \mathbb{R} \cup \{-\infty, +\infty\}$, its convex conjugate $\phi^\star : \mathbb{R} \to \mathbb{R} \cup \{-\infty, +\infty\}$ is defined as

$$\phi^\star(w) := \sup_{v \in \mathbb{R}} (vw - \phi(v)).$$

Since $(\phi^\star)^\star = \phi$, we have (Young-Fenchel inequality)

$$\phi(v) \geqslant vw - \phi^\star(w), \tag{5.8}$$

for any $w \in \mathrm{dom}(\phi^\star)$.

Lemma 5.10. *$\phi(v) = \frac{1}{\gamma}(v-1)^2$ and $\phi^\star(w) = w + \frac{\gamma}{4}w^2$ are convex conjugates.*

Proof of Lemma 5.10. By definition of the convex conjugate, we have

$$\begin{aligned}\phi^\star(w) &= \sup_{v\in\mathbb{R}}\left(-\frac{1}{\gamma}\cdot\left(v^2-(2+\gamma w)v+1\right)\right)\\ &= w+\frac{\gamma}{4}w^2,\end{aligned}$$

where the second line follows from plugging in the maximizer $v=\frac{\gamma w}{2}+1$. Note that the domain of $\phi^\star(w)$ is in fact $\mathbb{R}^d$ here. So, Eq. (5.8) holds for any $w\in\mathbb{R}^d$. □

We also introduce the concept of χ^2 divergence. For probability measures P and Q on the same measurable space $(\mathcal{A},\Omega)$ such that $Q\ll P$, the χ^2 divergence of Q from P is defined as

$$\chi^2(Q\parallel P):=\mathbb{E}_{a\sim P}\left[\left(\frac{dQ}{dP}(a)-1\right)^2\right],$$

where $\frac{dQ}{dP}(a)$ denotes the Radon-Nikodym derivative of Q with respect to P, which is a function mapping from a to $\mathbb{R}$.

Bounding the Decision-Estimation Coefficient.

We aim at bounding the Decision-Estimation Coefficient in this section. We use expression $\inf_{Q\in\mathcal{Q}_h}\mathbb{E}_{a^\star\sim Q}[f^\star(x,a^\star)]$ for $\mathsf{Smooth}_h(x)$. With this expression, we rewrite the Decision-Estimation Coefficient in the following: With respect to any context $x\in\mathcal{X}$ and estimator $\widehat{f}$ obtained from $\mathbf{Alg}_{\mathsf{Sq}}$, we denote

$$\begin{aligned}&\mathsf{dec}_\gamma(\mathcal{F};\widehat{f},x):=\\ &\inf_{P\in\Delta(\mathcal{A})}\sup_{Q\in\mathcal{Q}_h}\sup_{f\in\mathcal{F}}\mathbb{E}_{a\sim P,a^\star\sim Q}\left[f(x,a)-f(x,a^\star)-\frac{\gamma}{4}\cdot\left(\widehat{f}(x,a)-f(x,a)\right)^2\right],\end{aligned}$$

and define $\mathsf{dec}_\gamma(\mathcal{F}):=\sup_{\widehat{f},x}\mathsf{dec}_\gamma(\mathcal{F};\widehat{f},x)$ as the Decision-Estimation Coefficient. We remark here that $\sup_{Q\in\mathcal{Q}_h}\mathbb{E}_{a^\star\sim Q}[-f(x,a^\star)]=-\inf_{Q\in\mathcal{Q}_h}\mathbb{E}_{a\sim Q}[f^\star(x,a^\star)]$ so we are still compete with the best smoothing kernel within $\mathcal{Q}_h$.

We first state a result that helps eliminate the unknown f function in Decision-Estimation Coefficient (and thus the $\sup_{f\in\mathcal{F}}$ term), and bound Decision-Estimation Coefficient by the known $\widehat{f}$ estimator (from the regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$) and the χ^2-divergence from Q to P (whenever P and Q are probability measures).

Lemma 5.11. *Fix constant $\gamma > 0$ and context $x \in \mathcal{X}$. For any measures P and Q such that $Q \ll P$, we have*

$$\sup_{f\in\mathcal{F}} \mathbb{E}_{a\sim P, a^\star\sim Q}\left[f(x,a) - f(x,a^\star) - \frac{\gamma}{4}\cdot\left(\widehat{f}(x,a) - f(x,a)\right)^2\right]$$
$$\leqslant \mathbb{E}_{a\sim P}\left[\widehat{f}(x,a)\right] - \mathbb{E}_{a\sim Q}\left[\widehat{f}(x,a)\right] + \frac{1}{\gamma}\cdot\mathbb{E}_{a\sim P}\left[\left(\frac{dQ}{dP}(a) - 1\right)^2\right].$$

Proof of Lemma 5.11. We omit the dependence on the context $x \in \mathcal{X}$, and use abbreviations $f(a) := f(x,a)$ and $\widehat{f}(a) := \widehat{f}(x,a)$. Let $g := f - \widehat{f}$, we re-write the expression as

$$\begin{aligned}
&\sup_{f\in\mathcal{F}} \mathbb{E}_{a\sim P, a^\star\sim Q}\left[f(a) - f(a^\star) - \frac{\gamma}{4}\cdot\left(\widehat{f}(a) - f(a)\right)^2\right] \\
&= \sup_{g\in\mathcal{F}-\widehat{f}} \mathbb{E}_{a\sim P}\left[\widehat{f}(a)\right] - \mathbb{E}_{a^\star\sim Q}\left[\widehat{f}(a^\star)\right] - \mathbb{E}_{a^\star\sim Q}\left[g(a^\star)\right] + \mathbb{E}_{a\sim P}\left[g(a) - \frac{\gamma}{4}\cdot(g(a))^2\right] \\
&= \mathbb{E}_{a\sim P}\left[\widehat{f}(a)\right] - \mathbb{E}_{a\sim Q}\left[\widehat{f}(a)\right] \\
&\quad + \sup_{g\in\mathcal{F}-\widehat{f}}\left(\mathbb{E}_{a\sim Q}\left[-g(a)\right] - \mathbb{E}_{a\sim P}\left[(-g(a)) + \frac{\gamma}{4}\cdot(-g(a))^2\right]\right) \\
&= \mathbb{E}_{a\sim P}\left[\widehat{f}(a)\right] - \mathbb{E}_{a\sim Q}\left[\widehat{f}(a)\right] \\
&\quad + \sup_{g\in\mathcal{F}-\widehat{f}} \mathbb{E}_{a\sim P}\left[\frac{dQ}{dP}(a)\cdot(-g(a)) - \left((-g(a)) + \frac{\gamma}{4}\cdot(-g(a))^2\right)\right] \\
&= \mathbb{E}_{a\sim P}\left[\widehat{f}(a)\right] - \mathbb{E}_{a\sim Q}\left[\widehat{f}(a)\right] + \sup_{g\in\mathcal{F}-\widehat{f}} \mathbb{E}_{a\sim P}\left[\frac{dQ}{dP}(a)\cdot(-g(a)) - \phi^\star(-g(a))\right],
\end{aligned}$$

where we use the fact that $Q \ll P$ and $\phi^\star(w) = w + \frac{\gamma}{4}w^2$. Focus on the last term that depends on g takes the form of the RHS of Eq. (5.8): Consider $v = \frac{dQ}{dP}(a)$ and $w = -g(a)$ and apply Eq. (5.8) (with Lemma 5.10) eliminates the dependence on

g (since it works for any $w = -g(a)$) and leads to the following bound

$$\sup_{f \in \mathcal{F}} \mathbb{E}_{a \sim P, a^\star \sim Q}\left[f(a) - f(a^\star) - \frac{\gamma}{4} \cdot \left(\widehat{f}(a) - f(a)\right)^2\right]$$
$$\leqslant \mathbb{E}_{a \sim P}\left[\widehat{f}(a)\right] - \mathbb{E}_{a \sim Q}\left[\widehat{f}(a)\right] + \frac{1}{\gamma} \cdot \mathbb{E}_{a \sim P}\left[\left(\frac{dQ}{dP}(a) - 1\right)^2\right].$$

□

We now bound the Decision-Estimation Coefficient with sampling distribution defined in Eq. (5.2). We drop the dependence on t and define the sampling distribution in the generic form: Fix any constant $\gamma > 0$, context $x \in \mathcal{X}$ and estimator $\widehat{f}$, we define sampling distribution

$$P := M + (1 - M(\mathcal{A})) \cdot \mathbb{I}_{\widehat{a}}, \tag{5.9}$$

where $\widehat{a} := \arg\min_{a \in \mathcal{A}} \widehat{f}(x, a)$ and the measure M is defined through $M(\omega) := \int_{a \in \omega} m(a)\, d\mu(a)$ with

$$m(a) := \frac{1}{1 + h\gamma(\widehat{f}(x, a) - \widehat{f}(x, \widehat{a}))}. \tag{5.10}$$

Lemma 5.12. *Fix any constant $\gamma > 0$ and any set of regression function $\mathcal{F}$. Let P be the sampling distribution defined in Eq. (5.9), we then have $\mathsf{dec}_\gamma(\mathcal{F}) \leqslant \frac{2}{h\gamma}$.*

Proof of Lemma 5.12. As in the proof of Lemma 5.11, we omit the dependence on the context $x \in \mathcal{X}$ and use abbreviations $f(a) := f(x, a)$ and $\widehat{f}(a) := \widehat{f}(x, a)$.

We first notice that for any $Q \in \Omega_h$ we have $Q \ll M$ for M defined in Eq. (5.10): we have (i) $Q \ll \mu$ by definition, and (ii) $\mu \ll M$ (since $m(a) \geqslant \frac{1}{1+h\gamma} > 0$).[5] On the other side, however, we do not necessarily have $P \ll \mu$ for P defined in Eq. (5.9): It's possible to have $P(\{a^\star\}) > 0$ yet $\mu(\{a^\star\}) = 0$, e.g., μ is some continuous measure.

[5] We thus have $Q \ll P$ as well since P contains the component M by definition. We will, however, mostly be working with M due to its nice connection with the base measure μ, as defined in Eq. (5.10).

To isolate the corner case, we first give the following decomposition for any $Q \in \mathcal{Q}_h$ and $f \in \mathcal{F}$. With $P := M + (1 - M(\mathcal{A})) \cdot \mathbb{I}_{\widehat{a}}$, we have

$$
\begin{aligned}
&\mathbb{E}_{a\sim P, a^\star\sim Q}\left[f(a) - f(a^\star) - \frac{\gamma}{4}\cdot\left(\widehat{f}(a) - f(a)\right)^2\right] \\
&= (1 - M(\mathcal{A}))\cdot\left(f(\widehat{a}) - \frac{\gamma}{4}\cdot\left(\widehat{f}(\widehat{a}) - f(\widehat{a})\right)^2\right) \\
&\quad + \mathbb{E}_{a\sim M, a^\star\sim Q}\left[f(a) - f(a^\star) - \frac{\gamma}{4}\cdot\left(\widehat{f}(a) - f(a)\right)^2\right] \\
&= (1 - M(\mathcal{A}))\cdot\left(\widehat{f}(\widehat{a}) + \left(f(\widehat{a}) - \widehat{f}(\widehat{a})\right) - \frac{\gamma}{4}\cdot\left(\widehat{f}(\widehat{a}) - f(\widehat{a})\right)^2\right) \\
&\quad + \mathbb{E}_{a\sim M, a^\star\sim Q}\left[f(a) - f(a^\star) - \frac{\gamma}{4}\cdot\left(\widehat{f}(a) - f(a)\right)^2\right] \\
&\leqslant (1 - M(\mathcal{A}))\cdot\left(\widehat{f}(\widehat{a}) + \frac{1}{\gamma}\right) + \mathbb{E}_{a\sim M, a^\star\sim Q}\left[f(a) - f(a^\star) - \frac{\gamma}{4}\cdot\left(\widehat{f}(a) - f(a)\right)^2\right] \\
&\leqslant \frac{1 - M(\mathcal{A})}{\gamma} + (1 - M(\mathcal{A}))\cdot\widehat{f}(\widehat{a}) + \mathbb{E}_{a\sim M}\left[\widehat{f}(a)\right] - \mathbb{E}_{a\sim Q}\left[\widehat{f}(a)\right] \\
&\quad + \frac{1}{\gamma}\cdot\mathbb{E}_{a\sim M}\left[\left(\frac{dQ}{dM}(a) - 1\right)^2\right],
\end{aligned}
\tag{5.11}
$$

where the fourth line follows from applying AM-GM inequality and the fifth line follows from applying Lemma 5.11 with $Q \ll M$.[6] We now focus on the last four terms in Eq. (5.11). Denote $m(a) := \frac{dM}{d\mu}(a)$ and $q(a) := \frac{dQ}{d\mu}(a)$, with change of measures, we have

$$
\begin{aligned}
&(1 - M(\mathcal{A})\cdot\left(\widehat{f}(\widehat{a})\right) + \mathbb{E}_{a\sim M}\left[\widehat{f}(a)\right] - \mathbb{E}_{a\sim Q}\left[\widehat{f}(a)\right] + \frac{1}{\gamma}\cdot\mathbb{E}_{a\sim M}\left[\left(\frac{dQ}{dM}(a) - 1\right)^2\right] \\
&= \mathbb{E}_{a\sim\mu}\left[m(a)\cdot\left(\widehat{f}(a) - \widehat{f}(\widehat{a})\right)\right] - \mathbb{E}_{a\sim\mu}\left[q(a)\cdot\left(\widehat{f}(a) - \widehat{f}(\widehat{a})\right)\right] \\
&\quad + \frac{1}{\gamma}\cdot\mathbb{E}_{a\sim\mu}\left[m(a)\cdot\left(\frac{q(a)}{m(a)} - 1\right)^2\right] \\
&= \mathbb{E}_{a\sim\mu}\left[m(a)\cdot\left(\widehat{f}(a) - \widehat{f}(\widehat{a})\right)\right] - \mathbb{E}_{a\sim\mu}\left[q(a)\cdot\left(\widehat{f}(a) - \widehat{f}(\widehat{a})\right)\right]
\end{aligned}
$$

[6] With a slight abuse of notation, we use $\mathbb{E}_{a\sim M}[\cdot]$ denote the integration with respect to the sub-probability measure M.

$$
\begin{aligned}
&+\frac{1}{\gamma}\cdot\mathbb{E}_{a\sim\mu}\left[q(a)\cdot\frac{q(a)}{m(a)}-2q(a)+m(a)\right]\\
&=\mathbb{E}_{a\sim\mu}\left[m(a)\cdot\left(\widehat{f}(a)-\widehat{f}(\widehat{a})\right)\right]+\frac{1}{\gamma}\cdot\mathbb{E}_{a\sim Q}\left[\frac{q(a)}{m(a)}-\gamma\cdot\left(\widehat{f}(a)-\widehat{f}(\widehat{a})\right)\right]\\
&+\frac{M(\mathcal{A})-2}{\gamma}
\end{aligned}
\tag{5.12}
$$

Plugging Eq. (5.12) into Eq. (5.11) leads to

$$
\begin{aligned}
&\mathbb{E}_{a\sim P,a^\star\sim Q}\left[f(a)-f(a^\star)-\frac{\gamma}{4}\cdot\left(\widehat{f}(a)-f(a)\right)^2\right]\\
&\leqslant\mathbb{E}_{a\sim\mu}\left[m(a)\cdot\left(\widehat{f}(a)-\widehat{f}(\widehat{a})\right)\right]+\frac{1}{\gamma}\cdot\mathbb{E}_{a\sim Q}\left[\frac{q(a)}{m(a)}-\gamma\cdot\left(\widehat{f}(a)-\widehat{f}(\widehat{a})\right)\right]\\
&\leqslant\frac{2}{h\gamma},
\end{aligned}
\tag{5.13}
$$

where Eq. (5.13) follows from the fact that $m(a):=\frac{dM}{d\mu}(a)=\frac{1}{1+h\gamma(\widehat{f}(a)-\widehat{f}(\widehat{a}))}$ and $q(a):=\frac{dQ}{d\mu}(a)\leqslant\frac{1}{h}$ for any $Q\in\mathcal{Q}_h$. This certifies that $\mathrm{dec}_\gamma(\mathcal{F})\leqslant\frac{2}{h\gamma}$. □

5.8.1.2 Proof of Theorem 5.5

Theorem 5.5. *Fix any smoothness level $h\in(0,1]$. With an appropriate choice for $\gamma>0$, Algorithm 14 ensures that*

$$\mathbf{Reg}_{\mathsf{CB},h}(T)\leqslant\sqrt{4T\,\mathbf{Reg}_{\mathsf{Sq}}(T)/h},$$

with per-round runtime $O(\mathcal{T}_{\mathsf{Sq}}+\mathcal{T}_{\mathsf{Sample}})$ and maximum memory $O(\mathcal{M}_{\mathsf{Sq}}+\mathcal{M}_{\mathsf{Sample}})$.

Proof of Theorem 5.5. We use abbreviation $f_t(a):=f(x_t,a)$ for any $f\in\mathcal{F}$. Let $a_t^\star$ denote the action sampled according to the best smoothing kernel within $\mathcal{Q}_h$ (which could change from round to round). We let $\mathcal{E}$ denote the good event where the regret guarantee stated in Assumption 5.2 (i.e., $\mathbf{Reg}_{\mathsf{Sq}}(T):=\mathbf{Reg}_{\mathsf{Sq}}(T,T^{-1})$) holds with probability at least $1-T^{-1}$. Conditioned on this good event, following the analysis provided in Foster et al. (2020a), we decompose the contextual bandit

regret as follows.

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} f_t^\star(a_t) - f_t^\star(a_t^\star)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} f_t^\star(a_t) - f_t^\star(a_t^\star) - \frac{\gamma}{4}\cdot\left(\widehat{f}_t(a_t) - f_t^\star(a_t)\right)^2\right] \\
&\quad + \frac{\gamma}{4}\cdot\mathbb{E}\left[\sum_{t=1}^{T}\left(\widehat{f}_t(a_t) - f_t^\star(a_t)\right)^2\right] \\
&\leqslant T\cdot\frac{2}{h\gamma} + \frac{\gamma}{4}\cdot\mathbb{E}\left[\sum_{t=1}^{T}\left(\widehat{f}_t(a_t) - f_t^\star(a_t)\right)^2\right],
\end{aligned}$$

where the bound on the first term follows from Lemma 5.12. We analyze the second term below.

$$\begin{aligned}
&\frac{\gamma}{4}\cdot\mathbb{E}\left[\sum_{t=1}^{T}\left(\left(\widehat{f}_t(a_t) - \ell_t(a_t)\right)^2 - \left(f^\star(a_t) - \ell_t(a_t)\right)^2\right.\right. \\
&\quad \left.\left. + 2\left(\ell_t(a_t) - f_t^\star(a_t)\right)\cdot\left(\widehat{f}_t(a_t) - f_t^\star(a_t)\right)\right)\right] \\
&= \frac{\gamma}{4}\cdot\mathbb{E}\left[\sum_{t=1}^{T}\left(\left(\widehat{f}_t(a_t) - \ell_t(a_t)\right)^2 - \left(f_t^\star(a_t) - \ell_t(a_t)\right)^2\right)\right] \\
&\leqslant \frac{\gamma}{4}\cdot\mathbf{Reg}_{\mathrm{Sq}}(T),
\end{aligned}$$

where on the second line follows from the fact that $\mathbb{E}[\ell_t(a) \mid x_t] = f^\star(x_t, a)$ and ℓ_t is conditionally independent of a_t, and the third line follows from the bound on regression oracle stated in Assumption 5.2. As a result, we have

$$\mathbf{Reg}_{\mathrm{CB},h}(T) \leqslant \frac{2T}{h\gamma} + \frac{\gamma}{4}\cdot\mathbf{Reg}_{\mathrm{Sq}}(T) + O(1),$$

where the additional term $O(1)$ accounts for the expected regret suffered under event $\neg\mathcal{E}$. Taking $\gamma = \sqrt{8T/(h\cdot\mathbf{Reg}_{\mathrm{Sq}}(T))}$ leads to the desired result.

Computational complexity. We now discuss the computational complexity of Algorithm 14. At each round Algorithm 14 takes $O(1)$ calls to $\mathbf{Alg}_{\mathsf{Sq}}$ to obtain estimator $\widehat{f}_t$ and the best action $\widehat{a}_t$. Instead of directly form the action distribution defined in Eq. (5.2), Algorithm 14 uses Algorithm 15 to sample action $a_t \sim P_t$, which takes one call of the sampling oracle $\mathbf{Alg}_{\mathsf{Sample}}$ to draw a random action and $O(1)$ calls of the regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$ to compute the mean of the Bernoulli random variable. Altogether, Algorithm 14 has per-round runtime $O(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Sample}})$ and maximum memory $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Sample}})$. □

5.8.2 Proofs and Supporting Results for Section 5.4

This section is organized as follows. We first prove Proposition 5.7 in Section 5.8.2.1, then prove Theorem 5.8 in Section 5.8.2.2.

5.8.2.1 Proof of Proposition 5.7

The proof of Proposition 5.7 follows similar analysis as in Foster et al. (2020a), with minor changes to adapt to our settings.

Proposition 5.7. *For any* $b \in [B]$, *Algorithm 16 is* $\left(\frac{1}{2}, \sqrt{4T\,\mathbf{Reg}_{\mathsf{Sq}}(T)/h_b}\right)$*-stable, with per-round runtime* $O(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Sample}})$ *and maximum memory* $O(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Sample}})$.

Proof of Proposition 5.7. Fix the index $b \in [B]$ of the subroutine. We use shorthands $h = h_b$, $q_t = q_{t,b}$, $\rho_t = \rho_{t,b}$, $\gamma_t = \gamma_{t,b}$, and so forth. We also write $Z_t = Z_{t,b} := \mathbb{1}(I_t = b)$. Similar to the proof of Theorem 5.5, we use abbreviation $f_t(a) := f(x_t, a)$ for any $f \in \mathcal{F}$. Let $a_t^\star$ denote the action sampled according to the best smoothing kernel within $\mathcal{Q}_h$ (which could change from round to round).

We let $\mathcal{E}$ denote the good event where the regret guarantee stated in Assumption 5.3 (with $\mathbf{Reg}_{\mathsf{Sq}}(T) := \mathbf{Reg}_{\mathsf{Sq}}(T, T^{-1})$) holds with probability at least $1 - T^{-1}$. Conditioned on this good event, similar to the proof of Theorem 5.5 (and following

Foster et al. (2020a)), we decompose the contextual bandit regret as follows.

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\frac{Z_t}{q_t}(f_t^\star(a_t)-f_t^\star(a_t^\star))\right]\\
&=\mathbb{E}\left[\sum_{t=1}^{T}\frac{Z_t}{q_t}\left(f_t^\star(a_t)-f_t^\star(a_t^\star)-\frac{\gamma_t}{4}\cdot\left(\widehat{f}_t(a_t)-f_t^\star(a_t)\right)^2\right)\right]\\
&\quad+\mathbb{E}\left[\sum_{t=1}^{T}\frac{Z_t}{q_t}\cdot\frac{\gamma_t}{4}\cdot\left(\widehat{f}_t(a_t)-f_t^\star(a_t)\right)^2\right]\\
&\leqslant\mathbb{E}\left[\sum_{t=1}^{T}\frac{Z_t}{q_t}\cdot\frac{2}{h\gamma_t}\right]+\mathbb{E}\left[\sum_{t=1}^{T}\frac{Z_t}{q_t}\cdot\frac{\gamma_t}{4}\cdot\left(\widehat{f}_t(a_t)-f_t^\star(a_t)\right)^2\right]\\
&\leqslant\mathbb{E}\left[\max_{t\in[T]}\gamma_t^{-1}\right]\cdot\frac{2T}{h}+\mathbb{E}\left[\sum_{t=1}^{T}\frac{Z_t}{q_t}\cdot\frac{\gamma_t}{4}\cdot\left(\widehat{f}_t(a_t)-f_t^\star(a_t)\right)^2\right],
\end{aligned}
$$

where the bound on the first term follows from Lemma 5.12 (the third line, conditioned on Z_t). We bound the second term next.

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\frac{Z_t}{q_t}\cdot\frac{\gamma_t}{4}\cdot\left(\widehat{f}_t(a_t)-f_t^\star(a_t)\right)^2\right]\\
&=\frac{1}{4}\cdot\mathbb{E}\left[\sum_{t=1}^{T}\frac{Z_t}{q_t}\gamma_t\left(\left(\widehat{f}_t(a_t)-\ell_t(a_t)\right)^2-\left(f_t^\star(a_t)-\ell_t(a_t)\right)^2\right.\right.\\
&\quad\left.\left.+2\left(\ell_t(a_t)-f_t^\star(a_t)\right)\cdot\left(\widehat{f}_t(a_t)-f_t^\star(a_t)\right)\right)\right]\\
&=\frac{1}{4}\cdot\mathbb{E}\left[\sum_{t=1}^{T}\frac{Z_t}{q_t}\gamma_t\left((f_t(a_t)-\ell_t(a_t))^2-\left(f_t^\star(a_t)-\ell_t(a_t)\right)^2\right)\right]\\
&\leqslant\frac{1}{4}\cdot\mathbb{E}\left[\max_{t\in[T]}\frac{\gamma_t}{q_t}\right]\cdot\mathbf{Reg}_{\mathsf{Sq}}(T),
\end{aligned}
$$

where the last line follows from Assumption 5.3. As a result, we have

$$
\mathbf{Reg}_{\mathsf{Imp},h}(T)\leqslant\mathbb{E}\left[\max_{t\in[T]}\gamma_t^{-1}\right]\cdot\frac{2T}{h}+\frac{1}{4}\cdot\mathbb{E}\left[\max_{t\in[T]}\frac{\gamma_t}{q_t}\right]\cdot\mathbf{Reg}_{\mathsf{Sq}}(T)+O(1),
$$

where the additional $O(1)$ term is to account for the expected regret under event $\neg\mathcal{E}$. Notice that $\gamma_t := \sqrt{8T/(h \cdot \rho_t \cdot \mathbf{Reg}_{\mathsf{Sq}}(T))}$, which is non-increasing in t; and $\frac{\gamma_t}{q_t} \leqslant \gamma_t \rho_t$, which is non-decreasing in t. Thus, we have

$$\begin{aligned}\mathbf{Reg}_{\mathsf{Imp},h}(T) &\leqslant \mathbb{E}\left[\gamma_T^{-1}\right] \cdot \frac{2T}{h} + \frac{1}{4} \cdot \mathbb{E}[\gamma_T \rho_T] \cdot \mathbf{Reg}_{\mathsf{Sq}}(T) + O(1) \\ &= \mathbb{E}[\sqrt{\rho_T}] \cdot \sqrt{T\mathbf{Reg}_{\mathsf{Sq}}(T)/2h} + \mathbb{E}[\sqrt{\rho_T}]\sqrt{T\mathbf{Reg}_{\mathsf{Sq}}(T)/2h} + O(1) \\ &\leqslant \mathbb{E}[\sqrt{\rho_T}] \cdot \sqrt{4T\mathbf{Reg}_{\mathsf{Sq}}(T)/h}.\end{aligned}$$

Computational complexity. The computational compleity of Algorithm 16 can be analyzed in a similar way as the computational complexity of Algorithm 14, except with a *weighted* regression oracle $\mathbf{Alg}_{\mathsf{Sq}}$ this time. □

5.8.2.2 Proof of Theorem 5.8

We first restate the guarantee of CORRAL, specialized to our setting.

Theorem 5.13 (Agarwal et al. (2017))**.** *Fix an index $b \in [B]$. Suppose base algorithm b is $(\alpha_b, R_b(T))$-stable with respect to decision space indexed by b. If $\alpha_b < 1$, the CORRAL master algorithm, with learning rate $\eta > 0$, guarantees that*

$$\mathbb{E}\left[\sum_{t=1}^{T} f^\star(x_t, a_t) - \inf_{Q_t \in \mathcal{Q}_{h_b}} \mathbb{E}_{a_t^\star \sim Q_t}[f^\star(x_t, a_t^\star)]\right] = \widetilde{O}\left(\frac{B}{\eta} + T\eta + (R_b(T))^{\frac{1}{1-\alpha_b}} \eta^{\frac{\alpha_b}{1-\alpha_b}}\right).$$

Theorem 5.8. *Fix learning rate $\eta \in (0, 1]$, the CORRAL algorithm with Algorithm 16 as base algorithms guarantees that*

$$\mathbf{Reg}_{\mathsf{CB},h}(T) = \widetilde{O}\left(\frac{1}{\eta} + \frac{\eta\, T\, \mathbf{Reg}_{\mathsf{Sq}}(T)}{h}\right), \forall h \in (0, 1].$$

The CORRAL master algorithm has per-round runtime $\widetilde{O}(\mathcal{T}_{\mathsf{Sq}} + \mathcal{T}_{\mathsf{Sample}})$ and maximum memory $\widetilde{O}(\mathcal{M}_{\mathsf{Sq}} + \mathcal{M}_{\mathsf{Sample}})$.

Proof of Theorem 5.8. We prove the guarantee for any $h^\star \in [1/T, 1]$ as the otherwise the bound simply becomes vacuous. Recall that we initialize $B = \lceil \log T \rceil$ Algorithm 16 as base algorithms, each with a fixed smoothness parameter $h_b = 2^{-b}$, for $b \in [B]$. Using such geometric grid guarantees that there exists an $b^\star \in [B]$ such that $h_{b^\star} \leqslant h^\star \leqslant 2h_{b^\star}$. To obtain guarantee with respect to $h^\star$, it suffices to compete with subroutine $b^\star$ since $\mathcal{Q}_{h^\star} \subseteq \mathcal{Q}_{h_{b^\star}}$ by definition. Proposition 5.7 shows that the base algorithm indexed by $b^\star$ is $(\frac{1}{2}, \sqrt{4T\mathbf{Reg}_{\mathsf{Sq}}(T)/h_{b^\star}})$-stable. Plugging this result into Theorem 5.13 leads to the following guarantee:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} f^\star(x_t, a_t) - \inf_{Q_t \in \mathcal{Q}_{h^\star}} \mathbb{E}_{a_t^\star \sim Q_t}[f^\star(x_t, a_t^\star)]\right] \\
&\leqslant \mathbb{E}\left[\sum_{t=1}^{T} f^\star(x_t, a_t) - \inf_{Q_t \in \mathcal{Q}_{h_{b^\star}}} \mathbb{E}_{a_t^\star \sim Q_t}[f^\star(x_t, a_t^\star)]\right] \\
&= \widetilde{O}\left(\frac{B}{\eta} + T\eta + \frac{\eta\, T\, \mathbf{Reg}_{\mathsf{Sq}}(T)}{h_{b^\star}}\right) \\
&= \widetilde{O}\left(\frac{1}{\eta} + T\eta + \frac{\eta\, T\, \mathbf{Reg}_{\mathsf{Sq}}(T)}{h^\star}\right).
\end{aligned}$$

Computational complexity. The computational complexities (both runtime and memory) of the CORRAL master algorithm can be upper bounded by $\widetilde{O}(B \cdot \mathcal{C})$ where we use $\mathcal{C}$ denote the complexities of the base algorithms. We have $B = O(\log T)$ in our setting. Thus, directly plugging in the computational complexities of Algorithm 16 leads to the results. □

Recovering Adaptive Bounds in Krishnamurthy et al. (2020).

We discuss how our algorithms can also recover the adaptive regret bounds stated in Krishnamurthy et al. (2020) (Theorems 4 and 15), i.e.,

$$\mathbf{Reg}_{\mathsf{CB},h}(T) = \widetilde{O}\left(T^{\frac{1}{1+\beta}} (h^\star)^{-\beta} (\log|\mathcal{F}|)^{\frac{\beta}{1+\beta}}\right),$$

for any $h^\star \in (0, 1]$ and $\beta \in [0, 1]$. This line of analysis directly follows the proof

used in Krishnamurthy et al. (2020).

We focus on the case with $\mathbf{Reg}_{\mathsf{Sq}}(T) = O(\log(|\mathcal{F}|T))$. For base algorithm (Algorithm 16), following the analysis used in Krishnamurthy et al. (2020), we have

$$\begin{aligned}\mathbf{Reg}_{\mathsf{Imp},h}(T) &\leqslant \min\left\{T, \mathbb{E}[\sqrt{\rho_T}] \cdot \sqrt{4T\mathbf{Reg}_{\mathsf{Sq}}(T)/h}\right\} \\ &\leqslant \min\left\{T, \sqrt{\mathbb{E}[\rho_T]} \cdot \sqrt{4T\mathbf{Reg}_{\mathsf{Sq}}(T)/h}\right\} \\ &= O\left(T^{\frac{1}{1+\beta}} \cdot \left(\mathbb{E}[\rho_T]\mathbf{Reg}_{\mathsf{Sq}}(T)/h\right)^{\frac{\beta}{1+\beta}}\right),\end{aligned}$$

where on the first line we combine the regret obtained from Proposition 5.7 with a trivial upper bound T; on the second line we use the fact that $\sqrt{\cdot}$ is concave; and on the third line we use that fact that $\min\{A, B\} \leqslant A^{\gamma}B^{1-\gamma}$ for $A, B > 0$ and $\gamma \in [0,1]$ (taking $A = T$, $B = \sqrt{\mathbb{E}[\rho_T] \cdot 4T\mathbf{Reg}_{\mathsf{Sq}}(T)/h}$ and $\gamma = \frac{1-\beta}{1+\beta}$). This line of analysis thus shows that Algorithm 16 is $\left(\frac{\beta}{1+\beta}, \widetilde{O}\left(T^{\frac{1}{1+\beta}} \cdot \left(\mathbf{Reg}_{\mathsf{Sq}}(T)/h\right)^{\frac{\beta}{1+\beta}}\right)\right)$-stable for any $\beta \in [0,1]$.[7]

Now following the similar analysis as in the proof of Theorem 5.8, and consider $\mathbf{Reg}_{\mathsf{Sq}}(T) = O(\log(|\mathcal{F}|T))$ for the case with a finite set of regression functions, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} f^{\star}(x_t, a_t) - \inf_{Q_t \in \mathcal{Q}_{h^{\star}}} \mathbb{E}_{a_t^{\star} \sim Q_t}[f^{\star}(x_t, a_t^{\star})]\right] = \widetilde{O}\left(\frac{1}{\eta} + T\eta + T \cdot \left(\frac{\log(|\mathcal{F}|T)\,\eta}{h^{\star}}\right)^{\beta}\right),$$

for any $h^{\star} \in (0,1]$. Taking $\eta = T^{-\frac{1}{1+\beta}} \cdot (\log(|\mathcal{F}|T))^{-\frac{\beta}{1+\beta}}$ recovers the results presented in Krishnamurthy et al. (2020).

[7] As remarked in Krishnamurthy et al. (2020), the CORRAL algorithm works with both $\mathbb{E}[\rho_T^{\alpha}]$ and $(\mathbb{E}[\rho_T])^{\alpha}$.

Part III

Model Selection in Decision Making

6 BANDIT LEARNING WITH MULTIPLE BEST ARMS

We study a regret minimization problem with the existence of multiple best/near-optimal arms in the multi-armed bandit setting. We consider the case when the number of arms/actions is comparable or much larger than the time horizon, and make *no* assumptions about the structure of the bandit instance. Our goal is to design algorithms that can automatically adapt to the *unknown* hardness of the problem, i.e., the number of best arms. Our setting captures many modern applications of bandit algorithms where the action space is enormous and the information about the underlying instance/structure is unavailable. We first propose an adaptive algorithm that is agnostic to the hardness level and theoretically derive its regret bound. We then prove a lower bound for our problem setting, which indicates: (1) no algorithm can be minimax optimal simultaneously over all hardness levels; and (2) our algorithm achieves a rate function that is Pareto optimal. With additional knowledge of the expected reward of the best arm, we propose another adaptive algorithm that is minimax optimal, up to polylog factors, over *all* hardness levels. Experimental results confirm our theoretical guarantees and show advantages of our algorithms over the previous state-of-the-art.

6.1 Introduction

Multi-armed bandit problems describe exploration-exploitation trade-offs in sequential decision making. Most existing bandit algorithms tend to provide regret guarantees when the number of available arms/actions is smaller than the time horizon. In modern applications of bandit algorithm, however, the action space is usually comparable or even much larger than the allowed time horizon so that many existing bandit algorithms cannot even complete their initial exploration phases. Consider a problem of personalized recommendations, for example. For most users, the total number of movies, or even the amount of sub-categories, far exceeds the number of times they visit a recommendation site. Similarly, the enormous amount

of user-generated content on YouTube and Twitter makes it increasingly challenging to make optimal recommendations. The tension between a very large action space and a limited time horizon poses a realistic problem in which deploying algorithms that converge to an optimal solution over an asymptotically long time horizon *do not* give satisfying results. There is a need to design algorithms that can exploit the highest possible reward within a *limited* time horizon. Past work has partially addressed this challenge. The quantile regret proposed in Chaudhuri and Kalyanakrishnan (2018) to calculate regret with respect to an satisfactory action rather than the best one. The discounted regret analyzed in Ryzhov et al. (2012); Russo and Van Roy (2018) is used to emphasize short time horizon performance. Other existing works consider the extreme case when the number of actions is indeed infinite, and tackle such problems with one of two main assumptions: (1) the discovery of a near-optimal/best arm follows some probability measure with *known* parameters Berry et al. (1997); Wang et al. (2009); Aziz et al. (2018); Ghalme et al. (2020); (2) the existence of a *smooth* function represents the mean-payoff over a continuous subset Agrawal (1995); Kleinberg (2005); Kleinberg et al. (2008); Bubeck et al. (2011a); Locatelli and Carpentier (2018); Hadiji (2019). However, in many situations, neither assumption may be realistic. We make minimal assumptions in this chapter. We study the regret minimization problem over a time horizon T, which might be unknown, with respect to a bandit instance with n total arms, out of which m are best/near-optimal arms. We emphasize that the allowed time horizon and the given bandit instance should be viewed as features of *one* problem and together they indicate an intrinsic hardness level. We consider the case when the number of arms n is comparable or larger than the time horizon T so that no standard algorithm provides satisfying result. Our goal is to design algorithms that could adapt to the *unknown* m and achieve optimal regret.

6.1.1 Contributions and Organization

We make the following contributions. In Section 6.2, we formally define the regret minimization problem that represents the tension between a very large action space

and a limited time horizon; and capture the hardness level in terms of the number of best arms. We provide an adaptive algorithm that is agnostic to the *unknown* number of best arms in Section 6.3, and theoretically derive its regret bound. In Section 6.4, we prove a lower bound for our problem setting that indicates that there is no algorithm that can be optimal simultaneously over all hardness levels. Our lower bound also shows that our algorithm provided in Section 6.3 is Pareto optimal. With additional knowledge of the expected reward of the best arm, in Section 6.5, we provide an algorithm that achieves the non-adaptive minimax optimal regret, up to polylog factors, without the knowledge of the number of best arms. Experiments conducted in Section 6.6 confirm our theoretical guarantees and show advantages of our algorithms over previous state-of-the-art. We conclude this chapter in Section 6.7. Most of the proofs are deferred to the Appendix due to lack of space.

6.1.2 Additional Related Work

Time sensitivity and large action space. As bandit models are getting much more complex, usually with large or infinite action spaces, researchers have begun to pay attention to tradeoffs between regret and time horizons when deploying such models. Deshpande and Montanari (2012) study a linear bandit problem with ultra-high dimension, and provide algorithms that, under various assumptions, can achieve good reward within short time horizon. Russo and Van Roy (2018) also take time horizon into account and model time preference by analyzing a discounted regret. Chaudhuri and Kalyanakrishnan (2018) consider a quantile regret minimization problem where they define their regret with respect to expected reward ranked at $(1-\rho)$-th quantile. One could easily transfer their problem to our setting; however, their regret guarantee is sub-optimal. Katz-Samuels and Jamieson (2019); Aziz et al. (2018) also consider the problem with m best/near-optimal arms with no other assumptions, but they focus on the pure exploration setting; Aziz et al. (2018) additionally requires the knowledge of m. Another line of research considers the extreme case when the number arms is infinite, but with

some *known* regularities. Berry et al. (1997) proposes an algorithm with a minimax optimality guarantee under the situation where the reward of each arm follows *strictly* Bernoulli distribution; Teytaud et al. (2007) provides an anytime algorithm that works under the same assumption. Wang et al. (2009) relaxes the assumption on Bernoulli reward distribution, however, some other parameters are assumed to be known in their setting.

Continuum-armed bandit. Many papers also study bandit problems with continuous action spaces, where they embed each arm x into a bounded subset $\mathcal{X} \subseteq \mathbb{R}^d$ and assume there exists a smooth function f governing the mean-payoff for each arm. This setting is firstly introduced by Agrawal (1995). When the smoothness parameters are known to the learner or under various assumptions, there exists algorithms Kleinberg (2005); Kleinberg et al. (2008); Bubeck et al. (2011a) with near-optimal regret guarantees. When the smoothness parameters are unknown, however, Locatelli and Carpentier (2018) proves a lower bound indicating no strategy can be optimal simultaneously over all smoothness classes; under extra information, they provide adaptive algorithms with near-optimal regret guarantees. Although achieving optimal regret for all settings is impossible, Hadiji (2019) design adaptive algorithms and prove that they are Pareto optimal. Our algorithms are mainly inspired by the ones in Hadiji (2019); Locatelli and Carpentier (2018). A closely related line of work Valko et al. (2013); Grill et al. (2015); Bartlett et al. (2018); Shang et al. (2019) aims at minimizing simple regret in the continuum-armed bandit setting.

Adaptivity to unknown parameters. Bubeck et al. (2011b) argues the awareness of regularity is flawed and one should design algorithms that can *adapt* to the unknown environment. In situations where the goal is pure exploration or simple regret minimization, Katz-Samuels and Jamieson (2019); Valko et al. (2013); Grill et al. (2015); Bartlett et al. (2018); Shang et al. (2019) achieve near-optimal guarantees with unknown regularity because their objectives trade-off exploitation in favor of exploration. In the case of cumulative regret minimization, however, Locatelli and Carpentier (2018) shows no strategy can be optimal simultaneously over all smoothness classes. In special situations or under extra information, Bubeck

et al. (2011b); Bull et al. (2015); Locatelli and Carpentier (2018) provide algorithms that adapt in different ways. Hadiji (2019) borrows the concept of Pareto optimality from economics and provide algorithms with rate functions that are Pareto optimal. Adaptivity is studied in statistics as well: in some cases, only additional logarithmic factors are required Lepskii (1991); Birgé and Massart (1997); in others, however, there exists an additional polynomial cost of adaptation Cai et al. (2005).

6.2 Problem Setting

We consider the multi-armed bandit instance $\underline{\nu} = (\nu_1, \dots, \nu_n)$ with n probability distributions with means $\mu_i = \mathbb{E}_{X\sim\nu_i}[X] \in [0,1]$. Let $\mu_\star = \max_{i\in[n]}\{\mu_i\}$ be the highest mean and $S_\star = \{i \in [n] : \mu_i = \mu_\star\}$ denote the subset of best arms. The cardinality $|S_\star| = m$ is *unknown* to the learner. We could also generalize our setting to $S'_\star = \{i \in [n] : \mu_i \geqslant \mu_\star - \varepsilon(T)\}$ with unknown $|S'_\star|$ (i.e., situations where there is an unknown number of near-optimal arms). Setting ε to be dependent on T is to avoid an additive term linear in T, e.g., $\varepsilon \leqslant 1/\sqrt{T} \Rightarrow \varepsilon T \leqslant \sqrt{T}$. All theoretical results and algorithms presented in this chapter are applicable to this generalized setting with minor modifications. For ease of exposition, we focus on the case with multiple best arms throughout this chapter. At each time step $t \in [T]$, the algorithm/learner selects an action $A_t \in [n]$ and receives an independent reward $X_t \sim \nu_{A_t}$. We assume that $X_t - \mu_{A_t}$ is $(1/2)$-sub-Gaussian conditioned on A_t.[1] We measure the success of an algorithm through the expected cumulative (pseudo) regret:

$$R_T = T \cdot \mu_\star - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{A_t}\right].$$

We use $\mathcal{R}(T, n, m)$ to denote the set of regret minimization problems with allowed time horizon T and any bandit instance $\underline{\nu}$ with n total arms and m best

[1] We say a random variable X is σ-sub-Gaussian if $\mathbb{E}[\exp(\lambda X)] \leqslant \exp(\sigma^2\lambda^2/2)$ for all $\lambda \in \mathbb{R}$.

arms.[2] We emphasize that T is part of the problem instance. We are particularly interested in the case when n is comparable or even larger than T, which captures many modern applications where the available action space far exceeds the allowed time horizon. Although learning algorithms may not be able to pull each arm once, one should notice that the true/intrinsic hardness level of the problem could be viewed as n/m: selecting a subset uniformly at random with cardinality $\Theta(n/m)$ guarantees, with constant probability, the access to at least one best arm; but of course it is impossible to do this without knowing m. We quantify the *intrinsic* hardness level over a set of regret minimization problems $\mathcal{R}(T, n, m)$ as

$$\psi(\mathcal{R}(T, n, m)) = \inf\{\alpha \geqslant 0 : n/m \leqslant 2T^{\alpha}\},$$

where the constant 2 in front of T^{α} is added to avoid otherwise the trivial case with all best arms when the infimum is 0. $\psi(\mathcal{R}(T, n, m))$ is used here as it captures the *minimax* optimal regret over the set of regret minimization problem $\mathcal{R}(T, n, m)$, as explained later in our review of the MOSS algorithm and the lower bound. As smaller $\psi(\mathcal{R}(T, n, m))$ indicates easier problems, we then define the family of regret minimization problems with hardness level at most α as

$$\mathcal{H}_T(\alpha) = \{\cup\mathcal{R}(T, n, m) : \psi(\mathcal{R}(T, n, m)) \leqslant \alpha\},$$

with $\alpha \in [0, 1]$. Although T is necessary to define a regret minimization problem, we actually encode the hardness level into a single parameter α, which captures the *tension* between the complexity of bandit instance at hand and the allowed time horizon T: problems with different time horizons but the same α are equally difficult in terms of the achievable minimax regret (the exponent of T). We thus mainly study problems with T large enough so that we could mainly focus on the polynomial terms of T. We are interested in designing algorithms with *minimax* guarantees over $\mathcal{H}_T(\alpha)$, but *without* the knowledge of α.

[2] Our setting could be generalized to the case with infinite arms: one can consider embedding arms into an arm space $\mathcal{X}$ and let p be the probability that an arm sampled uniformly at random is (near-) optimal. $1/p$ will then serve a similar role as n/m does in the original definition.

MOSS and upper bound. In the classical setting, MOSS, proposed by Audibert and Bubeck (2009) and further generalized to the sub-Gaussian case Lattimore and Szepesvári (2020) and improved in terms of constant factors Garivier et al. (2018), achieves the minimax optimal regret. In this chapter, we will use MOSS as a subroutine with regret upper bound $O(\sqrt{nT})$ when $T \geqslant n$. For any problem in $\mathcal{H}_T(\alpha)$ with *known* α, one could run MOSS on a subset selected uniformly at random with cardinality $\widetilde{O}(T^\alpha)$ and achieve regret $\widetilde{O}(T^{(1+\alpha)/2})$.

Lower bound. The lower bound $\Omega(\sqrt{nT})$ in the classical setting does not work for our setting as its proof heavily relies on the existence of single best arm Lattimore and Szepesvári (2020). However, for problems in $\mathcal{H}_T(\alpha)$, we do have a matching lower bound $\Omega(T^{(1+\alpha)/2})$ as one could always apply the standard lower bound on an bandit instance with $n = \lfloor T^\alpha \rfloor$ and $m = 1$. For general value of m, a lower bound of the order $\Omega(\sqrt{T(n-m)/m}) = \Omega(T^{(1+\alpha)/2})$ for the m-best arms case could be obtained following similar analysis in Chapter 15 of Lattimore and Szepesvári (2020).

Although $\log T$ may appear in our bounds, throughout this chapter, we focus on problems with $T \geqslant 2$ as otherwise the bound is trivial.

6.3 An Adaptive Algorithm

Algorithm 17 takes time horizon T and a user-specified $\beta \in [1/2, 1)$ as input, and it is mainly inspired by Hadiji (2019). Algorithm 17 operates in iterations with geometrically-increasing length (roughly) $\Delta T_i = 2^{p+i}$ with $p = \lceil \log_2 T^\beta \rceil$. At each iteration i, it restarts MOSS on a set S_i consisting of $K_i = 2^{p+2-i}$ real arms selected uniformly at random *plus* a set of "virtual" *mixture-arms* (one from each of the $1 \leqslant j < i$ previous iterations, none if $i = 1$). The mixture-arms are constructed as follows. After each iteration i, let $\widehat{p}_i$ denote the vector of empirical sampling frequencies of the arms in that iteration (i.e., the k-th element of $\widehat{p}_i$ is the number of times arm k, including all previously constructed mixture-arms, was sampled in iteration i divided by the total number of samples ΔT_i). The mixture-arm for iteration i is the $\widehat{p}_i$-mixture of the arms, denoted by $\widetilde{\nu}_i$. When MOSS samples

from $\widetilde{\nu}_i$ it first draws $i_t \sim \widehat{p}_i$, then draws a sample from the corresponding arm ν_{i_t} (or $\widetilde{\nu}_{i_t}$). The mixture-arms provide a convenient summary of the information gained in the previous iterations, which is key to our theoretical analysis. Although our algorithm is working on fewer regular arms in later iterations, information summarized in mixture-arms is good enough to provide guarantees. We name our algorithm MOSS++ as it restarts MOSS at each iteration with past information summarized in mixture-arms. We provide an anytime version of Algorithm 17 in Section 6.8.1.2 via the standard doubling trick.

Algorithm 17 MOSS++

Input: Time horizon T and user-specified parameter $\beta \in [1/2, 1)$.

1: **Set:** $p = \lceil \log_2 T^\beta \rceil$, $K_i = 2^{p+2-i}$ and $\Delta T_i = \min\{2^{p+i}, T\}$.
2: **for** $i = 1, \ldots, p$ **do**
3: Run MOSS on a subset of arms S_i for ΔT_i rounds. S_i contains K_i real arms selected uniformly at random *and* the set of virtual mixture-arms from previous iterations, i.e., $\{\widetilde{\nu}_j\}_{j<i}$.
4: Construct a virtual mixture-arm $\widetilde{\nu}_i$ based on empirical sampling frequencies of MOSS above.

6.3.1 Analysis and Discussion

We use $\mu_S = \max_{\nu \in S}\{\mathbb{E}_{X\sim\nu}[X]\}$ to denote the highest expected reward over a set of distributions/arms S. For any algorithm that only works on S, we can decompose the regret into approximation error and learning error:

$$R_T = \underbrace{\mathbb{E}\left[T \cdot (\mu_\star - \mu_S)\right]}_{\text{expected approximation error due to the selection of } S} + \underbrace{\mathbb{E}\left[T \cdot \mu_S - \sum_{t=1}^{T} \mu_{A_t}\right]}_{\text{expected learning error due to the sampling rule } \{A_t\}_{t=1}^T}. \tag{6.1}$$

This type of regret decomposition was previously used in Kleinberg (2005); Auer et al. (2007); Hadiji (2019) to deal with the continuum-armed bandit problem.

We consider here a probabilistic version, with randomness in the selection of S, for the classical setting.

The main idea behind providing guarantees for MOSS++ is to decompose its regret at each iteration, using Eq. (6.1), and then bound the expected approximation error and learning error separately. The expected learning error at each iteration could always be controlled as $\widetilde{O}(T^{\beta})$ thanks to regret guarantees for MOSS and specifically chosen parameters p, K_i, ΔT_i. Let $i_\star$ be the largest integer such that $K_i \geqslant 2T^{\alpha} \log \sqrt{T}$ still holds. The expected approximation error in iteration $i \leqslant i_\star$ could be upper bounded by $\sqrt{T}$ following an analysis on hypergeometric distribution. As a result, the expected regret in iteration $i \leqslant i_\star$ is $\widetilde{O}(T^{\beta})$. Since the mixture-arm $\widetilde{\nu}_{i_\star}$ is included in all following iterations, we could further bound the expected approximation error in iteration $i > i_\star$ by $\widetilde{O}(T^{1+\alpha-\beta})$ after a careful analysis on $\Delta T_i / \Delta T_{i_\star}$. This intuition is formally stated and proved in Theorem 6.1.

Theorem 6.1. *Run MOSS++ with time horizon T and an user-specified parameter $\beta \in [1/2, 1)$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant C \left(\log_2 T\right)^{5/2} \cdot T^{\min\{\max\{\beta, 1+\alpha-\beta\}, 1\}},$$

where C is a universal constant.

Remark 6.2. *We primarily focus on the polynomial terms in T when deriving the bound, but put no effort in optimizing the polylog term. The $5/2$ exponent of $\log_2 T$ might be tightened as well.*

The theoretical guarantee is closely related to the user-specified parameter β: when $\beta > \alpha$, we suffer a multiplicative cost of adaptation $\widetilde{O}(T^{|(2\beta-\alpha-1)/2|})$, with $\beta = (1+\alpha)/2$ hitting the sweet spot, comparing to non-adaptive minimax regret; when $\beta \leqslant \alpha$, there is essentially no guarantees. One may hope to improve this result. However, our analysis in Section 6.4 indicates: (1) achieving minimax optimal regret for all settings simultaneously is *impossible*; and (2) the rate function achieved by MOSS++ is already *Pareto optimal*.

6.4 Lower Bound and Pareto Optimality

6.4.1 Lower Bound

In this section, we show that designing algorithms with the non-adaptive minimax optimal guarantee over all values of α is impossible. We first state the result in the following general theorem.

Theorem 6.3. *For any* $0 \leqslant \alpha' < \alpha \leqslant 1$, *assume* $T^{\alpha} \leqslant B$ *and* $\lfloor T^{\alpha} \rfloor - 1 \geqslant \max\{T^{\alpha}/4, 2\}$. *If an algorithm is such that* $\sup_{\omega \in \mathcal{H}_T(\alpha')} R_T \leqslant B$, *then the regret of this algorithm is lower bounded on* $\mathcal{H}_T(\alpha)$*:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \geqslant 2^{-10} T^{1+\alpha} B^{-1}. \tag{6.2}$$

To give an interpretation of Theorem 6.3, we consider any algorithm/policy π together with regret minimization problems $\mathcal{H}_T(\alpha')$ and $\mathcal{H}_T(\alpha)$ satisfying corresponding requirements. On one hand, if algorithm π achieves a regret that is order-wise larger than $\widetilde{O}(T^{(1+\alpha')/2})$ over $\mathcal{H}_T(\alpha')$, it is already not minimax optimal for $\mathcal{H}_T(\alpha')$. Now suppose π achieves a near-optimal regret, i.e., $\widetilde{O}(T^{(1+\alpha')/2})$, over $\mathcal{H}_T(\alpha')$; then, according to Eq. (6.2), π must incur a regret of order at least $\widetilde{\Omega}(T^{1/2+\alpha-\alpha'/2})$ on one problem in $\mathcal{H}_T(\alpha')$. This, on the other hand, makes algorithm π strictly sub-optimal over $\mathcal{H}_T(\alpha)$.

6.4.2 Pareto Optimality

We capture the performance of any algorithm by its dependence on polynomial terms of T in the asymptotic sense. Note that the hardness level of a problem is encoded in α.

Definition 6.4. *Let* $\theta : [0,1] \to [0,1]$ *denote a non-decreasing function. An algorithm achieves the rate function* θ *if*

$$\forall \varepsilon > 0, \forall \alpha \in [0,1], \quad \limsup_{T \to \infty} \frac{\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T}{T^{\theta(\alpha)+\varepsilon}} < +\infty.$$

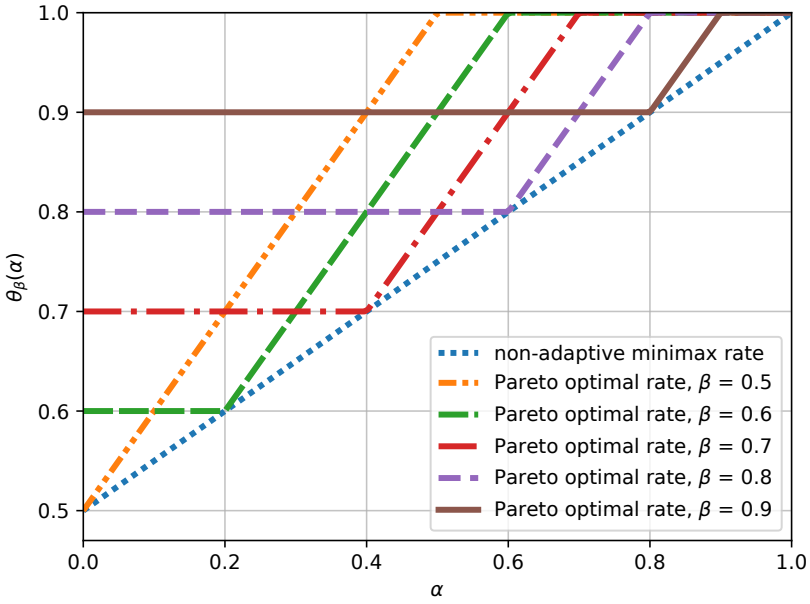


Figure 6.1: Pareto optimal rates for bandit learning with multiple best arms.

Recall that a function θ' is strictly smaller than another function θ in pointwise order if $\theta'(\alpha) \leqslant \theta(\alpha)$ for all α and $\theta'(\alpha_0) < \theta(\alpha_0)$ for at least one value of α_0. As there may not always exist a pointwise ordering over rate functions, following Hadiji (2019), we consider the notion of Pareto optimality over rate functions achieved by some algorithms.

Definition 6.5. *A rate function θ is Pareto optimal if it is achieved by an algorithm, and there is no other algorithm achieving a strictly smaller rate function θ' in pointwise order. An algorithm is Pareto optimal if it achieves a Pareto optimal rate function.*

Combining the results in Theorem 6.1 and Theorem 6.3 with above definitions, we could further obtain the following result in Theorem 6.6.

Theorem 6.6. *The rate function achieved by MOSS++ with any $\beta \in [1/2, 1)$, i.e.,*

$$\theta_\beta : \alpha \mapsto \min\{\max\{\beta, 1 + \alpha - \beta\}, 1\}, \tag{6.3}$$

is Pareto optimal.

Fig. 6.1 provides an illustration of the rate functions achieved by MOSS++ with different β as input, as well as the non-adaptive minimax optimal rate.

Remark 6.7. *One should notice that the naive algorithm running MOSS on a subset selected uniformly at random with cardinality $\widetilde{O}(T^{\beta'})$ is not Pareto optimal, since running MOSS++ with $\beta = (1+\beta')/2$ leads to a strictly smaller rate function. The algorithm provided in Chaudhuri and Kalyanakrishnan (2018), if transferred to our setting and allowing time horizon dependent quantile, is not Pareto optimal as well since it corresponds to the rate function $\theta(\alpha) = \max\{2.89\,\alpha, 0.674\}$.*

6.5 Learning with Extra Information

Although previous Section 6.4 gives negative results on designing algorithms that could optimally adapt to all settings, one could actually design such an algorithm *with* extra information. In this section, we provide an algorithm that takes the expected reward of the best arm $\mu_\star$ (or an estimated one with error up to $1/\sqrt{T}$) as extra information, and achieves near minimax optimal regret over all settings simultaneously. Our algorithm is mainly inspired by Locatelli and Carpentier (2018).

6.5.1 Algorithm

We name our Algorithm 19 Parallel as it maintains $\lceil \log T \rceil$ instances of subroutine, i.e., Algorithm 18, in parallel. Each subroutine SR_i is initialized with time horizon T and hardness level $\alpha_i = i/\lceil \log T \rceil$. We use $T_{i,t}$ to denote the number of samples allocated to SR_i up to time t, and represent its empirical regret at time t as $\widehat{R}_{i,t} = T_{i,t} \cdot \mu_\star - \sum_{t=1}^{T_{i,t}} X_{i,t}$ with $X_{i,t} \sim \nu_{A_{i,t}}$ being the t-th empirical reward obtained *by* SR_i and $A_{i,t}$ being the index of the t-th arm pulled *by* SR_i.

Parallel operates in iterations of length $\lceil \sqrt{T} \rceil$. At the beginning of each iteration, i.e., at time $t = i \cdot \lceil \sqrt{T} \rceil$ for $i \in \{0\} \cup [\lceil \sqrt{T} \rceil - 1]$, Parallel first selects the

Algorithm 18 MOSS Subroutine (SR)

Input: Time horizon T and hardness level α.

1: Select a subset of arms S_α uniformly at random with $|S_\alpha| = \lceil 2T^\alpha \log \sqrt{T} \rceil$ and run MOSS on S_α.

Algorithm 19 Parallel

Input: Time horizon T and the optimal reward $\mu_\star$.

1: **set:** $p = \lceil \log T \rceil$, $\Delta = \lceil \sqrt{T} \rceil$ and $t = 0$.
2: **for** $i = 1, \ldots, p$ **do**
3: Set $\alpha_i = i/p$, initialize SR_i with α_i, T; set $T_{i,t} = 0$, and $\widehat{R}_{i,t} = 0$.
4: **for** $i = 1, \ldots, \Delta - 1$ **do**
5: Select $k = \arg\min_{i \in [p]} \widehat{R}_{i,t}$ and run SR_k for Δ rounds.
6: Update $T_{k,t} = T_{k,t} + \Delta$, $\widehat{R}_{k,t} = T_{k,t} \cdot \mu_\star - \sum_{t=1}^{T_{k,t}} X_{k,t}$, $t = t + \Delta$.

subroutine with the lowest (breaking ties arbitrarily) empirical regret so far, i.e., $k = \arg\min_{i \in [\lceil \log T \rceil]} \widehat{R}_{i,t}$; it then *resumes* the learning process of SR_k, from where it halted, for another $\lceil \sqrt{T} \rceil$ more pulls. All the information is updated at the end of that iteration. An anytime version of Algorithm 19 is provided in Section 6.8.3.3.

6.5.2 Analysis

As Parallel discretizes the hardness parameter over a grid with interval $1/\lceil \log T \rceil$, we first show that running the best subroutine alone leads to regret $\widetilde{O}(T^{(1+\alpha)/2})$.

Lemma 6.8. *Suppose α is the true hardness parameter and $\alpha_i - 1/\lceil \log T \rceil < \alpha \leqslant \alpha_i$, run Algorithm 18 with time horizon T and α_i leads to the following regret bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant C \log T \cdot T^{(1+\alpha)/2},$$

where C is a universal constant.

Since Parallel always allocates new samples to the subroutine with the lowest empirical regret so far, we know that the regret of every subroutine should be roughly of the same order at time T. In particular, all subroutines should achieve

regret $\widetilde{O}(T^{(1+\alpha)/2})$, as the best subroutine does. Parallel then achieves the non-adaptive minimax optimal regret, up to polylog factors, *without* knowing the true hardness level α.

Theorem 6.9. *For any $\alpha \in [0,1]$ unknown to the learner, run Parallel with time horizon T and optimal expected reward $\mu_\star$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant C\ (\log T)^2\, T^{(1+\alpha)/2},$$

where C is a universal constant.

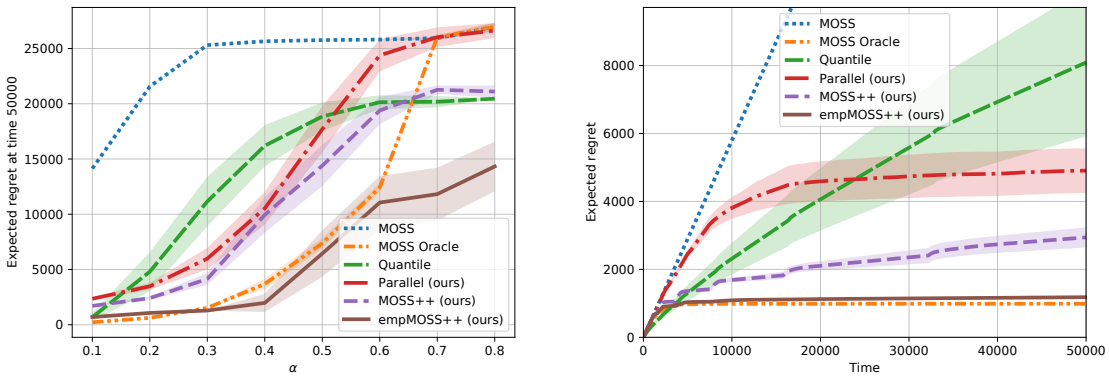
6.6 Experiments

We conduct three experiments to compare our algorithms with baselines. In Section 6.6.1, we compare the performance of each algorithm on problems with varying hardness levels. We examine how the regret curve of each algorithm increases on synthetic and real-world datasets in Section 6.6.2 and Section 6.6.3, respectively.

We first introduce the nomenclature of the algorithms. We use MOSS to denote the standard MOSS algorithm; and MOSS Oracle to denote Algorithm 18 with *known* α. Quantile represents the algorithm (QRM2) proposed by Chaudhuri and Kalyanakrishnan (2018) to minimize the regret with respect to the $(1-\rho)$-th quantile of means among arms, without the knowledge of ρ. One could easily transfer Quantile to our settings with top-ρ fraction of arms treated as best arms. As suggested in Chaudhuri and Kalyanakrishnan (2018), we reuse the statistics obtained in previous iterations of Quantile to improve its sample efficiency. We use MOSS++ to represent the vanilla version of Algorithm 17; and use empMOSS++ to represent an empirical version such that: (1) empMOSS++ reuse statistics obtained in previous round, as did in Quantile; and (2) instead of selecting K_i real arms uniformly at random at the i-th iteration, empMOSS++ selects K_i arms with the highest empirical mean for $i > 1$. We choose $\beta = 0.5$ for MOSS++ and empMOSS++

in all experiments.[3] All results are averaged over 100 experiments. Shaded area represents 0.5 standard deviation for each algorithm.

6.6.1 Adaptivity to Hardness Level



(a) (b)

Figure 6.2: Experiments on synthetic dataset. (a) Comparison of regret with varying hardness level α (b) Comparison of progressive regret curve with $\alpha = 0.25$.

We compare our algorithms with baselines on regret minimization problems with different hardness levels. For this experiment, we generate best arms with expected reward 0.9 and sub-optimal arms with expected reward evenly distributed among $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. All arms follow Bernoulli distribution. We set the time horizon to $T = 50000$ and consider the total number of arms $n = 20000$. We vary α from 0.1 to 0.8 (with interval 0.1) to control the number of best arms $m = \lceil n/2T^{\alpha} \rceil$ and thus the hardness level. In Fig. 6.2(a), the regret of any algorithm gets larger as α increases, which is expected. MOSS does not provide satisfying performance due to the large action space and the relatively small time horizon. Although implemented in an anytime fashion, Quantile could be roughly viewed as an algorithm that runs MOSS on a subset selected uniformly at random with cardinality $T^{0.347}$. Quantile

[3] Increasing β generally leads to worse performance on problems with small α but better performance on problems with large α.

displays good performance when $\alpha = 0.1$, but suffers regret much worse than MOSS++ and empMOSS++ when α gets larger. Note that the regret curve of Quantile gets flattened at 20000 is expected: it simply learns the best sub-optimal arm and suffers a regret $50000 \times (0.9 - 0.5)$. Although Parallel enjoys near minimax optimal regret, the regret it suffers from is the summation of 11 subroutines, which hurts its empirical performance. empMOSS++ achieves performance comparable to MOSS Oracle when α is small, and achieve the best empirical performance when $\alpha \geqslant 0.3$. When $\alpha \geqslant 0.7$, MOSS Oracle needs to explore most/all of the arms to statistically guarantee the finding of at least one best arm, which hurts its empirical performance.

6.6.2 Comparison of Progressive Regret Curve

We compare how the regret curve of each algorithm increases in Fig. 6.2(b). We consider the same regret minimization configurations as described in Section 6.6.1 with $\alpha = 0.25$. empMOSS++, MOSS++ and Parallel all outperform Quantile with empMOSS++ achieving the performance closest to MOSS Oracle. MOSS Oracle, Parallel and empMOSS++ have flattened their regret curve indicating they could confidently recommend the best arm. The regret curves of MOSS++ and Quantile do not flat as the random-sampling component in each of their iterations encourage them to explore new arms. Comparing to MOSS++, Quantile keeps increasing its regret at a much faster rate and with a much larger variance, which empirically confirms the sub-optimality of their regret guarantees.

6.6.3 Real-World Dataset

We also compare all algorithms in a realistic setting of recommending funny captions to website visitors. We use a real-world dataset from the *New Yorker Magazine* Cartoon Caption Contest[4]. The dataset of 1-3 star caption ratings/rewards for Contest 652 consists of $n = 10025$ captions[5]. We use the ratings to compute Bernoulli

[4] https://www.newyorker.com/cartoons/contest.

[5] Available online at https://nextml.github.io/caption-contest-data.

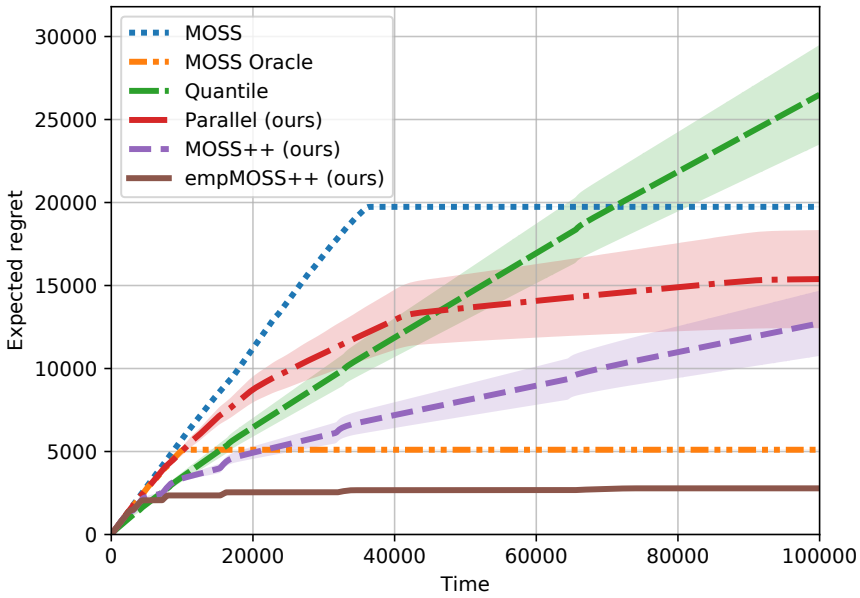


Figure 6.3: Comparison of progressive regret curve on a real-world dataset from the *New Yorker Magazine* Cartoon Caption Contest.

reward distributions for each caption as follows. The mean of each caption/arm i is calculated as the percentage p_i of its ratings that were funny or somewhat funny (i.e., 2 or 3 stars). We normalize each p_i with the best one and then threshold each: if $p_i \geqslant 0.8$, then put $p_i = 1$; otherwise leave p_i unaltered. This produces a set of $m = 54$ best arms with rewards 1 and all other 9971 arms with rewards among $[0, 0.8]$. We set $T = 10^5$ and this results in a hardness level around $\alpha \approx 0.43$.

Using these Bernoulli reward models, we compare the performance of each algorithm, as shown in Fig. 6.3. MOSS, MOSS Oracle, Parallel and empMOSS++ have flattened their regret curve indicating they could confidently recommend the funny captions (i.e., best arms). Although MOSS could eventually identify a best arm in this problem, its cumulative regret is more than 7x of the regret achieved by empMOSS++ due to its initial exploration phase. The performance of Quantile is even worse, and its cumulative regret is more than 9x of the regret

achieved by empMOSS++. One surprising phenomenon is that empMOSS++ outperforms MOSS Oracle in this realistic setting. Our hypothesis is that MOSS Oracle is a little bit conservative and selects an initial set with cardinality too large. This experiment demonstrates the effectiveness of empMOSS++ and MOSS++ in modern applications of bandit algorithm with large action space and limited time horizon.

6.7 Discussion

We study a regret minimization problem with large action space but limited time horizon, which captures many modern applications of bandit algorithms. Depending on the number of best/near-optimal arms, we encode the hardness level, in terms of minimax regret achievable, of the given regret minimization problem into a single parameter α, and we design algorithms that could adapt to this *unknown* hardness level. Our first algorithm MOSS++ takes a user-specified parameter β as input and provides guarantees as long as $\alpha < \beta$; our lower bound further indicates the rate function achieved by MOSS++ is Pareto optimal. Although no algorithm can achieve near minimax optimal regret over all α simultaneously, as demonstrated by our lower bound, we overcome this limitation with an (often) easily-obtained extra information and propose Parallel that is near-optimal for all settings. Inspired by MOSS++, We also propose empMOSS++ with excellent empirical performance. Experiments on both synthetic and real-world datasets demonstrate the efficiency of our algorithms over the previous state-of-the-art.

6.8 Proofs and Supporting Results

6.8.1 Proofs and Supporting Results for Section 6.3

We introduce the notation $R_{T|\mathcal{F}} = T \cdot \mu_\star - \mathbb{E}[\sum_{t=1}^{T} X_t|\mathcal{F}]$ for any σ-algebra $\mathcal{F}$. One should also notice that $\mathbb{E}[R_{T|\mathcal{F}}] = R_T$.

6.8.1.1 Proof of Theorem 6.1

Lemma 6.10. *For an instance with n total arms and m best arms, and for a subset S selected uniformly at random with cardinality k, the probability that none of the best arms are selected in S is upper bounded by $\exp(-mk/n)$.*

Proof. Consider selecting k items out of n items without replacement; and suppose there are m target items. Let $\mathcal{E}$ denote the event where none of the target items are selected, we then have

$$\begin{aligned}
\mathbb{P}(\mathcal{E}) &= \frac{\binom{n-m}{k}}{\binom{n}{k}} = \frac{\frac{(n-m)!}{(n-m-k)!k!}}{\frac{n!}{(n-k)!k!}} \\
&= \frac{(n-m)!}{(n-m-k)!} \cdot \frac{(n-k)!}{n!} \\
&= \prod_{i=0}^{k-1} \frac{n-m-i}{n-i} \\
&\leqslant \left(\frac{n-m}{n}\right)^k \qquad (6.4) \\
&\leqslant \exp\left(-\frac{m}{n} \cdot k\right), \qquad (6.5)
\end{aligned}$$

where Eq. (6.4) comes from the fact that $\frac{n-m-i}{n-i}$ is decreasing in i; and Eq. (6.5) comes from the fact that $1 - x \leqslant \exp(-x)$ for all $x \in \mathbb{R}$.

Selecting arms with replacement gives the same guarantee (which directly goes to Eq. (6.4)), and can be used in corner cases when $k > n$. □

Theorem 6.1. *Run MOSS++ with time horizon T and an user-specified parameter $\beta \in [1/2, 1)$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant C (\log_2 T)^{5/2} \cdot T^{\min\{\max\{\beta, 1+\alpha-\beta\}, 1\}},$$

where C is a universal constant.

Proof. Let $T_i = \sum_{j=1}^{i} \Delta T_j$. We first notice that Algorithm 17 is a valid algorithm

in the sense that it selects an arm A_t for any $t \in [T]$, i.e., it does not terminate before time T: the argument is clearly true if there exists $i \in [p]$ such that $\Delta T_i = T$; otherwise, we can show that

$$T_p = \sum_{i=1}^{p} \Delta T_i = 2(2^{2p} - 1) \geqslant 2^{2p} \geqslant T,$$

for all $\beta \in [1/2, 1)$.

We will only consider the case when $\alpha < \beta$ in the following since otherwise Theorem 6.1 trivially holds due to $T^{1+\alpha-\beta} \geqslant T$.

Let $\mathcal{F}_{i-1}$ represents information collected up to the beginning of iteration i, including the random selection of S_i. We use $\mu_{S_i|\mathcal{F}_i} = \max_{\nu \in S_i}\{\mathbb{E}_{X\sim\nu}[X|\mathcal{F}_{i-1}]\}$ to denote the maximum expected reward among arms in S_i conditioned on $\mathcal{F}_{i-1}$. We use $R_{\Delta T_i|\mathcal{F}_{i-1}} = \Delta T_i \cdot \mu_\star - \mathbb{E}[\sum_{t=T_{i-1}+1}^{T_i} X_t|\mathcal{F}_{i-1}]$ to denote the conditional expected cumulative regret at iteration i; and further have $R_{\Delta T_i} = \mathbb{E}[R_{\Delta T_i|\mathcal{F}_{i-1}}]$.

For any virtual mixture-arm $\widetilde{\nu}_j$ created before iteration i (i.e., $j < i$), we use $\widetilde{\mu}_{j|\mathcal{F}_j} = \mathbb{E}_{X\sim\widetilde{\nu}_j}[X|\mathcal{F}_j]$ to denote its expected reward conditioned on $\mathcal{F}_j$. Conditioning on $\mathcal{F}_j$, let X be a sample from a virtual mixture-arm $\widetilde{\nu}_i$, which is realized by first sampling an index j_t (of a real arm) from the empirical measure, and then draw X from the real arm ν_{j_t}. We then know that $X - \widetilde{\mu}_{j|\mathcal{F}_j}$ is (conditional) $(\sqrt{2}/2)$-sub-Gaussian: $X - \widetilde{\mu}_{j|\mathcal{F}_j} = (X - \mu_{j_t}) + \left(\mu_{j_t} - \widetilde{\mu}_{j|\mathcal{F}_j}\right)$ and thus for any $\lambda \in \mathbb{R}$,

$$\begin{aligned}
\mathbb{E}\left[\exp\left(\lambda\left(X - \widetilde{\mu}_{j|\mathcal{F}_j}\right)\right)\middle|\mathcal{F}_j\right] &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(\lambda\left(X - \widetilde{\mu}_{j|\mathcal{F}_j}\right)\right)|j_t\right]\middle|\mathcal{F}_j\right] \\
&= \mathbb{E}\left[\exp\left(\lambda(\mu_{j_t} - \widetilde{\mu}_{j|\mathcal{F}_j})\right)\mathbb{E}\left[\exp\left(\lambda\left(X - \mu_{j_t}\right)\right)|j_t\right]\middle|\mathcal{F}_j\right] \\
&\leqslant \exp\left(\frac{\lambda^2/4}{2}\right)\mathbb{E}\left[\exp\left(\lambda(\mu_{j_t} - \widetilde{\mu}_{j|\mathcal{F}_j})\right)\middle|\mathcal{F}_j\right] \\
&\leqslant \exp\left(\frac{\lambda^2/4}{2} + \frac{\lambda^2/4}{2}\right) \qquad (6.6) \\
&= \exp\left(\frac{\lambda^2/2}{2}\right)
\end{aligned}$$

where Eq. (6.6) comes from the fact that $\mu_{j_t} \in [0, 1]$ and $\mathbb{E}[\mu_{j_t}|\mathcal{F}_j] = \widetilde{\mu}_{j|\mathcal{F}_j}$. In the

following, we'll directly plug in the regret bound of MOSS for the 1-sub-Gaussian case.

Applying Eq. (6.1) on $R_{\Delta T_i|\mathcal{F}_{i-1}}$ leads to

$$R_{\Delta T_i|\mathcal{F}_{i-1}} = \Delta T_i \cdot \left(\mu_\star - \mu_{S_i|\mathcal{F}_{i-1}}\right) + \left(\Delta T_i \cdot \mu_{S_i|\mathcal{F}_{i-1}} - \mathbb{E}\left[\sum_{t=T_{i-1}+1}^{T_i} \mu_{A_t} \,\middle|\, \mathcal{F}_{i-1}\right]\right), \tag{6.7}$$

where, by a slightly abuse of notations, we use μ_{A_t} to refer to the mean of arm $A_t \in S_i$, which could also be the mean of a virtual arm constructed in one of the previous iterations.

We first consider the learning error for any iteration $i \in [p]$. $\mu_{S_i|\mathcal{F}_{i-1}}$ is measurable with respect to $\mathcal{F}_{i-1}$ and thus can be thought as fixed at time $T_{i-1}+1$ (conditioned on $\mathcal{F}_{i-1}$). Since MOSS restarts at each iteration, conditioning on the information available at the beginning of the i-th iteration, i.e., $\mathcal{F}_{i-1}$, and apply the regret bound for MOSS, we have:

$$\begin{aligned}
\Delta T_i \cdot \mu_{S_i|\mathcal{F}_{i-1}} - \mathbb{E}\left[\sum_{t=T_{i-1}+1}^{T_i} \mu_{A_t} \,\middle|\, \mathcal{F}_{i-1}\right] &\leqslant 39\sqrt{|S_i|\Delta T_i} + |S_i| && (6.8)\\
&= 39\sqrt{(K_i + i - 1)\Delta T_i} + (K_i + i - 1)\\
&\leqslant 39\sqrt{K_i \Delta T_i + (p-1)\Delta T_i} + (K_i + p - 1) && (6.9)\\
&\leqslant 39\sqrt{2^{2p+2} + (p-1)T} + 2^{p+1} + (p-1) && (6.10)\\
&\leqslant 39\sqrt{16T^{2\beta} + \log_2(T^\beta)\, T} + 4T^\beta + \log_2 T^\beta && (6.11)\\
&\leqslant 166\,(\log_2 T)^{1/2} \cdot T^\beta, && (6.12)
\end{aligned}$$

where Eq. (6.8) comes from the guarantee of MOSS Lattimore and Szepesvári (2020); Eq. (6.9) comes from $i \leqslant p$; Eq. (6.10) comes from the definition of K_i and

ΔT_i; Eq. (6.11) comes from the fact that $p = \lceil \log_2 T^\beta \rceil \leqslant \log_2 T^\beta + 1$; Eq. (6.12) comes from some trivial boundings on the constant.[6]

Taking expectation over randomness in $\mathcal{F}_{i-1}$ in Eq. (6.7), we obtain

$$R_{\Delta T_i} \leqslant \Delta T_i \cdot \mathbb{E}\left[(\mu_\star - \mu_{S_i|\mathcal{F}_{i-1}})\right] + 166\,(\log_2 T)^{1/2} \cdot T^\beta. \tag{6.13}$$

Now, we only need to consider the first term, i.e., the expected approximation error over the i-th iteration. Let $\mathcal{E}_i$ denote the event that none of the best arms, among regular arms, is selected in S_i, according to Lemma 6.10, we further have

$$\begin{aligned} \Delta T_i \cdot \mathbb{E}\left[(\mu_\star - \mu_{S_i|\mathcal{F}_{i-1}})\right] &\leqslant \Delta T_i \cdot (0 \cdot \mathbb{P}(\neg\mathcal{E}_i) + 1 \cdot \mathbb{P}(\mathcal{E}_i)) && (6.14)\\ &\leqslant \Delta T_i \cdot \exp(-K_i/(2T^\alpha)), && (6.15) \end{aligned}$$

where we use the fact the $\mu_i \in [0,1]$ in Eq. (6.14); and directly plug $n/m \leqslant 2T^\alpha$ into Eq. (6.5) to get Eq. (6.15).

Let $i_\star \in [p]$ be the largest integer, if exists, such that $K_{i_\star} \geqslant 2T^\alpha \log\sqrt{T}$, we then have that, for any $i \leqslant i_\star$,

$$\Delta T_i \cdot \mathbb{E}\left[(\mu_\star - \mu_{S_i|\mathcal{F}_{i-1}})\right] \leqslant \Delta T_i/\sqrt{T} \leqslant T/\sqrt{T} \leqslant \sqrt{T}. \tag{6.16}$$

Note that this choice of $i_\star$ indicates $T^\alpha \log T \leqslant K_{i_\star} < 2T^\alpha \log T$.

If we have $K_1 < 2T^\alpha \log\sqrt{T}$, we then set $i_\star = 1$. Notice that $K_1 = 2^{p+1} = 2^{\lceil \log_2 T^\beta \rceil + 1} \geqslant 2T^\beta > 2T^\alpha$, we then have

$$\Delta T_1 \cdot \mathbb{E}\left[(\mu_\star - \mu_{S_1|\mathcal{F}_0})\right] \leqslant \Delta T_1 \exp(-1) \leqslant 2^{p+1}\exp(-1) < 2T^\beta. \tag{6.17}$$

Combining Eq. (6.13) with Eq. (6.16) or Eq. (6.17), we have for any $i \leqslant i_\star$, and in particular for $i = i_\star$,

$$R_{\Delta T_i} \leqslant \max\{\sqrt{T}, 2T^\beta\} + 166\,(\log_2 T)^{1/2} \cdot T^\beta$$

[6] One can remove the $(\log_2 T)^{1/2}$ term in many cases, e.g., when $\beta > 1/2$ and T is large enough (with respect to β). However, we mainly focus on the polynomial terms here.

$$\leqslant 168\,(\log_2 T)^{1/2}\cdot T^{\beta}. \tag{6.18}$$

In the case when $i_\star = p$ or when $\Delta T_{i_\star} = \min\{2^{p+i}, T\} = T$, we know that MOSS++ will in fact stop at a time step no larger than $T_{i_\star}$ (since the allowed time horizon is T), and incur no regret in iterations $i > i_\star$. In the following, we only consider the case when $i_\star < p$ and $\Delta T_{i_\star} = 2^{p+i_\star}$. As a result, we have $K_{i_\star}\Delta T_{i_\star} = 2^{2p+2}$ and thus

$$\Delta T_{i_\star} = \frac{2^{2p+2}}{K_{i_\star}} > \frac{2^{2p+1}}{T^{\alpha}\log T}, \tag{6.19}$$

where Eq. (6.19) comes from the fact that $K_{i_\star} < \max\{2T^{\alpha}\log T, 2T^{\alpha}\log\sqrt{T}\} = 2T^{\alpha}\log T$ by definition of $i_\star$.

We now analysis the expected approximation error for iteration $i > i_\star$. Since the sampling information during the $i_\star$-th iteration is summarized in the virtual mixture-arm $\widetilde{\nu}_{i_\star}$, and being added to all S_i for all $i > i_\star$. Recall that $\widetilde{\mu}_{i_\star|\mathcal{F}_{i_\star}} = \mathbb{E}_{X\sim\widetilde{\nu}_{i_\star}}[X|\mathcal{F}_{i_\star}]$ denotes the expected reward of sampling according to the virtual mixture-arm $\widetilde{\nu}_{i_\star}$, conditioned on information collected in $\mathcal{F}_{i_\star}$. For any $i > i_\star$, we then have

$$\begin{aligned}
\Delta T_i\cdot\mathbb{E}\left[(\mu_\star - \mu_{S_i|\mathcal{F}_{i-1}})\right] &\leqslant \Delta T_i\cdot\mathbb{E}[(\mu_\star - \widetilde{\mu}_{i_\star|\mathcal{F}_{i_\star}})] \\
&= \frac{\Delta T_i}{\Delta T_{i_\star}}\cdot\mathbb{E}[\Delta T_{i_\star}\cdot(\mu_\star - \widetilde{\mu}_{i_\star|\mathcal{F}_{i_\star}})] \\
&= \frac{\Delta T_i}{\Delta T_{i_\star}}\cdot\mathbb{E}\left[\left(\Delta T_{i_\star}\cdot\mu_\star - \sum_{t=T_{i_\star-1}+1}^{T_{i_\star}}\mu_{A_t}\right]\right) \\
&= \frac{\Delta T_i}{\Delta T_{i_\star}}\cdot R_{\Delta T_{i_\star}} \\
&< \frac{\Delta T_i}{\frac{2^{2p+1}}{T^{\alpha}\log T}}\cdot 168\,(\log_2 T)^{1/2}\cdot T^{\beta} \\
&\leqslant \frac{T^{1+\alpha+\beta}}{2^{2p}}\cdot 84\,(\log_2 T)^{3/2} \qquad (6.20)\\
&\leqslant 84\,(\log_2 T)^{3/2}\cdot T^{1+\alpha-\beta}, \qquad (6.21)
\end{aligned}$$

where Eq. (6.20) comes from the fact that $\Delta T_i \leqslant T$ and some rewriting; Eq. (6.21) comes from the fact that $p = \lceil \log_2 T^\beta \rceil \geqslant \log_2 T^\beta$.

Combining Eq. (6.21) and Eq. (6.13) gives the following regret bound for iterations $i > i_\star$:

$$R_{\Delta T_i} \leqslant 250 \, (\log_2 T)^{3/2} \cdot T^{\max\{\beta, 1+\alpha-\beta\}},$$

where the constant 250 simply comes from $84 + 166$.

Since the cumulative regret is non-decreasing in t, we have

$$\begin{aligned} R_T &\leqslant \sum_{i=1}^{p} R_{\Delta T_i} \\ &\leqslant 250 \, p \, (\log_2 T)^{3/2} \cdot T^{\max\{\beta, 1+\alpha-\beta\}} \\ &\leqslant 250 \, (\log_2 T + 1) \cdot (\log_2 T)^{3/2} \cdot T^{\max\{\beta, 1+\alpha-\beta\}} \qquad (6.22) \\ &\leqslant 251 \, (\log_2 T)^{5/2} \cdot T^{\max\{\beta, 1+\alpha-\beta\}}, \end{aligned}$$

where Eq. (6.22) comes from the fact that $p = \lceil \log_2(T^\beta) \rceil \leqslant \log_2(T^\beta) + 1 \leqslant \log_2 T + 1$. Our results follows after noticing that $R_T \leqslant T$ is a trivial upper bound. □

6.8.1.2 Anytime Version

Algorithm 20 Anytime version of MOSS++

Input: User specified parameter $\beta \in [1/2, 1)$.

1: **for** $i = 0, 1, \ldots$ **do**
2: Run Algorithm 17 with parameter β for 2^i rounds (note that we will set $p = \lceil \log_2 2^{i\beta} \rceil = \lceil i\beta \rceil$).

Corollary 6.11. *For any unknown time horizon T, run Algorithm 20 with an user-specified parameter $\beta \in [1/2, 1)$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant C \, (\log_2 T)^{5/2} \cdot T^{\min\{\max\{\beta, 1+\alpha-\beta\}, 1\}},$$

where C *is a universal constant.*

Proof. Let $t_\star$ be the smallest integer such that

$$\sum_{i=0}^{t_\star} 2^i = 2^{t_\star+1} - 1 \geqslant T.$$

We then only need to run Algorithm 17 for at most $t_\star$ times. By the definition of $t_\star$, we also know that $2^{t_\star} \leqslant T$, which leads to $t_\star \leqslant \log_2 T$.

Let $\gamma = \min\{\max\{\beta, 1+\alpha-\beta\}, 1\}$. From Theorem 6.1 we know that the regret at $i \in [t_\star]$-th round, denoted as R_{2^i}, could be upper bounded by

$$\begin{aligned} R_{2^i} &\leqslant 251\,(\log_2 2^i)^{5/2} \cdot (2^i)^\gamma \\ &= 251\, i^{5/2} \cdot (2^\gamma)^i \\ &\leqslant 251\, t_\star^{5/2} \cdot (2^\gamma)^i \\ &\leqslant 251\,(\log_2 T)^{5/2} \cdot (2^\gamma)^i. \end{aligned}$$

For $i = 0$, we have $R_{2^0} \leqslant 1 \leqslant 251\,(\log_2 T)^{5/2} \cdot (2^\gamma)^0$ as well as long as $T \geqslant 2$.

Now for the unknown time horizon T, we could upper bound the regret by

$$\begin{aligned} R_T &\leqslant \sum_{i=0}^{t_\star} R_{2^i} \\ &\leqslant 251\,(\log_2 T)^{5/2} \cdot \left(\sum_{i=0}^{t_\star} (2^\gamma)^i \right) \\ &\leqslant 251\,(\log_2 T)^{5/2} \cdot \int_{x=0}^{t_\star+1} (2^\gamma)^x \mathrm{d}x \qquad (6.23) \\ &= 251\,(\log_2 T)^{5/2} \cdot \frac{1}{\log 2^\gamma} \cdot \left((2^\gamma)^{t_\star+1} - 1 \right) \\ &\leqslant \frac{2^\gamma}{\gamma \log 2} 251\,(\log_2 T)^{5/2} \cdot T^\gamma \\ &\leqslant 1449\,(\log_2 T)^{5/2} \cdot T^\gamma, \qquad (6.24) \end{aligned}$$

where Eq. (6.23) comes from upper bounding summation by integral; and Eq. (6.24) comes from a trivial bound on the constant when $1/2 \leqslant \gamma \leqslant 1$. □

6.8.2 Proofs and Supporting Results for Section 6.4

6.8.2.1 Proof of Theorem 6.3

Theorem 7.1. *For any $0 \leqslant \alpha' < \alpha \leqslant 1$, assume $T^{\alpha} \leqslant B$ and $\lfloor T^{\alpha} \rfloor - 1 \geqslant \max\{T^{\alpha}/4, 2\}$. If an algorithm is such that $\sup_{\omega \in \mathcal{H}_T(\alpha')} R_T \leqslant B$, then the regret of this algorithm is lower bounded on $\mathcal{H}_T(\alpha)$:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \geqslant 2^{-10} T^{1+\alpha} B^{-1}. \tag{7.1}$$

The proof of Theorem 6.3 is mainly inspired by the proofs of lower bounds in Locatelli and Carpentier (2018); Hadiji (2019). Before the start of the proof, we first state a generalized version of Pinsker's inequality developed in Hadiji (2019) (Lemma 3 therein).

Lemma 6.12. *Let $\mathbb{P}$ and $\mathbb{Q}$ be two probability measures. For any random variable $Z \in [0,1]$, we have*

$$|\mathbb{E}_{\mathbb{P}}[Z] - \mathbb{E}_{\mathbb{Q}}[Z]| \leqslant \sqrt{KL(\mathbb{P}, \mathbb{Q})/2}.$$

We consider $K+1$ bandit instances $\{\underline{\nu}_i\}_{i=0}^{K}$ such that each bandit instance is a collection of n distributions $\underline{\nu}_i = (\nu_{i1}, \nu_{i2}, \ldots, \nu_{in})$ where each ν_{ij} represents a Gaussian distribution $\mathcal{N}(\mu_{ij}, 1/4)$ with $\mu_{ij} = \mathbb{E}[\nu_{ij}]$. For any given $0 \leqslant \alpha' < \alpha \leqslant 1$ and time horizon T large enough, we choose $n, m_0, m, K \in \mathbb{N}_+$ such that the following three conditions are satisfied:

1. $n = m_0 + Km$;
2. $n/m_0 \leqslant 2T^{\alpha'}$;
3. $n/m \leqslant 2T^{\alpha}$.

Proposition 6.13. *Integers satisfying the above three conditions exist. For instance, we could first fix* $m \in \mathbb{N}_+$ *and set* $K = \lfloor T^\alpha \rfloor - 1 \geqslant 2$.[7] *One could then set* $m_0 = m\lceil T^{\alpha-\alpha'} \rceil$ *and* $n = m_0 + Km$.

Proof. We notice that the first condition holds by construction. We now show that the second and the third conditions hold.

For the second condition, we have

$$\begin{aligned}
\frac{n}{m_0} &= \frac{m_0 + Km}{m_0} \\
&= 1 + \frac{m(\lfloor T^\alpha \rfloor - 1)}{m\lceil T^{\alpha-\alpha'} \rceil} \\
&\leqslant 1 + \frac{T^\alpha}{T^{\alpha-\alpha'}} \\
&\leqslant 2T^{\alpha'}.
\end{aligned}$$

For the third condition, we have

$$\begin{aligned}
\frac{n}{m} &= \frac{m_0 + Km}{m} \\
&= \frac{m\lceil T^{\alpha-\alpha'} \rceil + (\lfloor T^\alpha \rfloor - 1)m}{m} \\
&= \lceil T^{\alpha-\alpha'} \rceil + \lfloor T^\alpha \rfloor - 1 \\
&= \left(\lceil T^{\alpha-\alpha'} \rceil - 1\right) + \lfloor T^\alpha \rfloor \\
&\leqslant T^{\alpha-\alpha'} + T^\alpha \\
&\leqslant 2T^\alpha.
\end{aligned}$$

□

Now we group n distribution into $K+1$ different groups based on their indices: $S_0 = [m_0]$ and $S_i = [m_0 + i \cdot m] \backslash [m_0 + (i-1) \cdot m]$. Let $\Delta \in (0, 1]$ be a parameter to be tuned later, we then define $K+1$ bandit instances $\underline{\nu}_i$ for $i \in \{0\} \cup [K]$ by assigning

[7] $K \geqslant 2$ holds for T large enough.

different values to their means μ_{ij}:

$$\mu_{ij} = \begin{cases} \Delta/2 & \text{if } j \in S_0, \\ \Delta & \text{if } j \in S_i \text{ and } i \neq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{6.25}$$

We could clearly see there are m_0 best arms in instance $\underline{\nu}_0$ and m best arms in instances $\underline{\nu}_i, \forall i \in [K]$. Based on our construction in Proposition 6.13, we could then conclude that, with time horizon T, the regret minimization problem with respect to $\underline{\nu}_0$ is in $\mathcal{H}_T(\alpha')$; and similarly the regret minimization problem with respect to $\underline{\nu}_i$ is in $\mathcal{H}_T(\alpha), \forall i \in [K]$.

For any $t \in [T]$, the tuple of random variables $H_t = (A_1, X_1, \ldots, A_t, X_t)$ is the outcome of an algorithm interacting with an bandit instance up to time t. Let $\Omega_t = ([n] \times \mathbb{R})^t \subseteq \mathbb{R}^{2t}$ and $\mathcal{F}_t = \mathfrak{B}(\Omega_t)$; one could then define a measurable space $(\Omega_t, \mathcal{F}_t)$ for H_t. The random variables $A_1, X_1, \ldots, A_t, X_t$ that make up the outcome are defined by their coordinate projections:

$$A_t(a_1, x_1, \ldots, a_t, x_t) = a_t \quad \text{and} \quad X_t(a_1, x_1, \ldots, a_t, x_t) = x_t.$$

For any fixed algorithm/policy π and bandit instance $\underline{\nu}_i$, $\forall i \in \{0\} \cup [K]$, we are now constructing a probability measure $\mathbb{P}_{i,t}$ over $(\Omega_t, \mathcal{F}_t)$. Note that a policy π is a sequence $(\pi_t)_{t=1}^T$, where π_t is a probability kernel from $(\Omega_{t-1}, \mathcal{F}_{t-1})$ to $([n], 2^{[n]})$. For each i, we define another probability kernel $p_{i,t}$ from $(\Omega_{t-1} \times [n], \mathcal{F}_{t-1} \otimes 2^{[n]})$ to $(\mathbb{R}, \mathfrak{B}(\mathbb{R}))$ that models the reward. Assuming the reward is distributed according to $\mathcal{N}(\mu_{ia_t}, 1/4)$, we give its explicit expression for any $B \in \mathfrak{B}(\mathbb{R})$ as:

$$p_{i,t}\big((a_1, x_1, \ldots, a_t), B\big) = \int_B \sqrt{\frac{2}{\pi}} \exp\big(-2(x - \mu_{ia_t})\big)\, dx.$$

The probability measure over $\mathbb{P}_{i,t}$ over $(\Omega_t, \mathcal{F}_t)$ could then be define recursively as $\mathbb{P}_{i,t} = p_{i,t}\big(\pi_t \mathbb{P}_{i,t-1}\big)$. We use $\mathbb{E}_i$ to denote the expectation taken with respect to $\mathbb{P}_{i,T}$. Apply the same analysis as on page 21 of Hadiji (2019), we obtain the following

proposition on KL decomposition.

Proposition 6.14.

$$\mathrm{KL}\left(\mathbb{P}_{0,T}, \mathbb{P}_{i,T}\right) = \mathbb{E}_0\left[\sum_{t=1}^{T} \mathrm{KL}\left(\mathcal{N}(\mu_{0A_t}, 1/4), \mathcal{N}\left(\mu_{iA_t}, 1/4\right)\right)\right].$$

With respect to notations and constructions described above, we now prove Theorem 6.3.

Proof. (Theorem 6.3) Let $N_{S_i}(T) = \sum_{t=1}^{T} \mathbb{1}\left(A_t \in S_i\right)$ denote the number of times the algorithm π selects an arm in S_i up to time T. Let $R_{i,T}$ denote the expected (pseudo) regret achieved by the algorithm π interacting with the bandit instance $\underline{\nu}_i$. Based on the construction of bandit instance in Eq. (6.25), we have

$$R_{0,T} \geqslant \frac{\Delta}{2} \sum_{i=1}^{K} \mathbb{E}_0\left[N_{S_i}(T)\right], \tag{6.26}$$

and $\forall i \in [K]$,

$$R_{i,T} \geqslant \frac{\Delta}{2}\left(T - \mathbb{E}_i[N_{S_i}(T)]\right) = \frac{T\Delta}{2}\left(1 - \frac{\mathbb{E}_i[N_{S_i}(T)]}{T}\right). \tag{6.27}$$

According to Proposition 6.14 and the calculation of KL-divergence between two Gaussian distributions, we further have

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}_{0,T}, \mathbb{P}_{i,T}) &= \mathbb{E}_0\left[\sum_{t=1}^{T} \mathrm{KL}\left(\mathcal{N}(\mu_{0A_t}, 1/4), \mathcal{N}\left(\mu_{iA_t}, 1/4\right)\right)\right] \\
&= \mathbb{E}_0\left[\sum_{t=1}^{T} 2\left(\mu_{0A_t} - \mu_{iA_t}\right)^2\right] \\
&= 2\mathbb{E}_0\left[N_{S_i}(T)\right]\Delta^2,
\end{aligned} \tag{6.28}$$

where Eq. (6.28) comes from the fact that μ_{0j} and μ_{ij} only differs for $j \in S_i$ and the difference is exactly Δ.

We now consider the average regret over $i \in [K]$:

$$\begin{aligned}
\frac{1}{K}\sum_{i=1}^{K} R_{i,T} &\geqslant \frac{T\Delta}{2}\left(1 - \frac{1}{K}\sum_{i=1}^{K}\frac{\mathbb{E}_i[N_{S_i}(T)]}{T}\right) \\
&\geqslant \frac{T\Delta}{2}\left(1 - \frac{1}{K}\sum_{i=1}^{K}\left(\frac{\mathbb{E}_0[N_{S_i}(T)]}{T} + \sqrt{\frac{\mathrm{KL}(\mathbb{P}_{0,T}, \mathbb{P}_{i,T})}{2}}\right)\right) && (6.29) \\
&= \frac{T\Delta}{2}\left(1 - \frac{1}{K}\frac{\sum_{i=1}^{K}\mathbb{E}_0[N_{S_i}(T)]}{T} - \frac{1}{K}\sum_{i=1}^{K}\sqrt{\mathbb{E}_0\left[N_{S_i}(T)\right]\Delta^2}\right) && (6.30) \\
&\geqslant \frac{T\Delta}{2}\left(1 - \frac{1}{K} - \sqrt{\frac{\sum_{i=1}^{K}\mathbb{E}_0\left[N_{S_i}(T)\right]\Delta^2}{K}}\right) && (6.31) \\
&\geqslant \frac{T\Delta}{2}\left(1 - \frac{1}{K} - \sqrt{\frac{2\Delta R_{0,T}}{K}}\right) && (6.32) \\
&\geqslant \frac{T\Delta}{2}\left(\frac{1}{2} - \sqrt{\frac{2\Delta B}{K}}\right), && (6.33)
\end{aligned}$$

where Eq. (6.29) comes from applying Lemma 6.12 with $Z = N_{S_i}(T)/T$ and $\mathbb{P} = \mathbb{P}_{0,T}$ and $\mathbb{Q} = \mathbb{P}_{i,T}$; Eq. (6.30) comes from applying Eq. (6.28); Eq. (6.31) comes from concavity of $\sqrt{\cdot}$ and the fact that $\sum_{i=1}^{K}\mathbb{E}_0[N_{S_i}(T)] \leqslant T$; Eq. (6.32) comes from applying Eq. (6.26); and finally Eq. (6.33) comes from the fact that $K \geqslant 2$ by construction and the assumption that $R_{0,T} \leqslant B$.

To obtain a large value for Eq. (6.33), one could maximize Δ while still make $\sqrt{2\Delta B/K} \leqslant 1/4$. Set $\Delta = 2^{-5}KB^{-1}$, following Eq. (6.33), we obtain

$$\begin{aligned}
\frac{1}{K}\sum_{i=1}^{K} R_{i,T} &\geqslant 2^{-8}TKB^{-1} \\
&= 2^{-8}T\left(\lfloor T^{\alpha}\rfloor - 1\right)B^{-1} && (6.34) \\
&\geqslant 2^{-10}T^{1+\alpha}B^{-1}, && (6.35)
\end{aligned}$$

where Eq. (6.34) comes from the construction of K; and Eq. (6.35) comes from the assumption that $\lfloor T^{\alpha}\rfloor - 1 \geqslant T^{\alpha}/4$.

Now we only need to make sure $\Delta = 2^{-5}KB^{-1} \leqslant 1$. Since we have $K = \lfloor T^{\alpha} \rfloor - 1 \leqslant T^{\alpha}$ by construction and $T^{\alpha} \leqslant B$ by assumption, we obtain $\Delta = 2^{-5}KB^{-1} \leqslant 2^{-5} < 1$ as desired. □

6.8.2.2 Proof of Theorem 6.6

Lemma 6.15. *Suppose an algorithm achieves rate function θ, then for any $0 < \alpha \leqslant \theta(0)$, we have*

$$\theta(\alpha) \geqslant 1 + \alpha - \theta(0). \tag{6.36}$$

Proof. Fix $0 < \alpha \leqslant \theta(0)$. For any $\varepsilon > 0$, there exists constant c_1 and c_2 such that for sufficiently large T,

$$\sup_{\omega \in \mathcal{H}_T(0)} R_T \leqslant c_1 T^{\theta(0)+\varepsilon} \quad \text{and} \quad \sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant c_2 T^{\theta(\alpha)+\varepsilon}.$$

Let $B = \max\{c_1, 1\} \cdot T^{\theta(0)+\varepsilon}$, we could see that $T^{\alpha} \leqslant T^{\theta(0)} \leqslant B$ holds by assumption. For T large enough, the condition $\lfloor T^{\alpha} \rfloor - 1 \geqslant \max\{T^{\alpha}/4, 2\}$ of Theorem 6.3 holds. We then have

$$c_2 T^{\theta(\alpha)+\varepsilon} \geqslant 2^{-10} T^{1+\alpha} \left(\max\{c_1, 1\} \cdot T^{\theta(0)+\varepsilon}\right)^{-1} = 2^{-10} T^{1+\alpha-\theta(0)-\varepsilon} / \max\{c_1, 1\}.$$

For T sufficiently large, we then must have

$$\theta(\alpha) + \varepsilon \geqslant 1 + \alpha - \theta(0) - \varepsilon.$$

Let $\varepsilon \to 0$ leads to the desired result. □

Lemma 6.16. *Suppose a rate function θ is achieved by an algorithm, then we must have*

$$\theta(\alpha) \geqslant \min\{\max\{\theta(0), 1 + \alpha - \theta(0)\}, 1\}, \tag{6.37}$$

with $\theta(0) \in [1/2, 1]$.

Proof. For any rate function θ achieved by an algorithm, we first notice that $\theta(\alpha) \geqslant \theta(\alpha')$ for any $0 \leqslant \alpha' < \alpha \leqslant 1$ since $\mathcal{H}_T(\alpha') \subseteq \mathcal{H}_T(\alpha)$; this also implies $\theta(\alpha) \geqslant \theta(0)$. From Lemma 6.15, we further obtain $\theta(\alpha) \geqslant 1 + \alpha - \theta(0)$ if $\alpha \leqslant \theta(0)$. Thus, for any $\alpha \in (0, \theta(0)]$, we have

$$\theta(\alpha) \geqslant \max\{\theta(0), 1 + \alpha - \theta(0)\}. \tag{6.38}$$

Note that this indicates $\theta(\theta(0)) = 1$, as we trivially have $R_T \leqslant T$. For any $\alpha \in (\theta(0), 1]$, we have $\theta(\alpha) \geqslant \theta(\theta(0)) = 1$, which leads to $\theta(\alpha) = 1$ for $\alpha \in [\theta(0), 1]$. To summarize, we obtain the desired result in Eq. (6.37). We have $\theta(0) \in [1/2, 1]$ since the minimax optimal rate among problems in $\mathcal{H}_T(0)$ is $1/2$. □

Theorem 7.8. *The rate function achieved by MOSS++ with any $\beta \in [1/2, 1)$, i.e.,*

$$\theta_\beta : \alpha \mapsto \min\{\max\{\beta, 1 + \alpha - \beta\}, 1\}, \tag{7.4}$$

is Pareto optimal.

Proof. From Theorem 6.1, we know that the rate in Eq. (6.3) is achieved by Algorithm 17 with input β. We only need to prove that no other algorithms achieve strictly smaller rates in pointwise order.

Suppose, by contradiction, we have θ' achieved by an algorithm such that $\theta'(\alpha) \leqslant \theta_\beta(\alpha)$ for all $\alpha \in [0, 1]$ and $\theta'(\alpha_0) < \theta(\alpha_0)$ for at least one $\alpha_0 \in [0, 1]$. We then must have $\theta'(0) \leqslant \theta_\beta(0) = \beta$. We consider the following two exclusive cases.

Case 1: $\theta'(0) = \beta$**.** According to Lemma 6.16, we must have $\theta' \geqslant \theta_\beta$, which leads to a contradiction.

Case 2: $\theta'(0) = \beta' < \beta$**.** According Lemma 6.16, we must have $\theta' \geqslant \theta_{\beta'}$. However, $\theta_{\beta'}$ is not strictly better than θ_β, e.g., $\theta_{\beta'}(2\beta - 1) = 2\beta - \beta' > \beta = \theta_\beta(2\beta - 1)$, which also leads to a contradiction. □

6.8.3 Proofs and Supporting Results for Section 6.5

6.8.3.1 Proof of Lemma 6.8

Lemma 6.8. *Suppose α is the true hardness parameter and $\alpha_i - 1/\lceil \log T \rceil < \alpha \leqslant \alpha_i$, run Algorithm 18 with time horizon T and α_i leads to the following regret bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant C \log T \cdot T^{(1+\alpha)/2},$$

where C is a universal constant.

Proof. Let $\mathcal{E}$ denote the event that none of the best arm is selected in S_{α_i}. According to Lemma 6.10, the definition of α and the assumption that $\alpha \leqslant \alpha_i$, we know that $\mathbb{P}(\mathcal{E}) \leqslant 1/\sqrt{T}$. We now upper bound the regret:

$$\begin{aligned}
R_T &\leqslant \left(39\sqrt{|S_{\alpha_i}|T} + |S_{\alpha_i}|\right) \cdot \mathbb{P}(\neg\mathcal{E}) + T \cdot \mathbb{P}(\mathcal{E}) && (6.39)\\
&\leqslant \left(39\sqrt{|S_{\alpha_i}|T} + |S_{\alpha_i}|\right) \cdot 1 + T \cdot \frac{1}{\sqrt{T}} \\
&\leqslant 56\,(\log T)^{1/2} \cdot T^{(1+\alpha_i)/2} + 2\log T \cdot T^{\alpha_i} + \sqrt{T} \\
&\leqslant 59\,\log T \cdot T^{(1+\alpha_i)/2} \\
&< 59\,\log T \cdot T^{(1+\alpha)/2} \cdot T^{1/(2\lceil \log T \rceil)} && (6.40)\\
&\leqslant 59\sqrt{e}\,\log T \cdot T^{(1+\alpha)/2}, && (6.41)
\end{aligned}$$

where Eq. (6.39) comes from the regret bound of MOSS; Eq. (6.40) comes from the assumption that $\alpha_i < \alpha + 1/\lceil \log T \rceil$; and Eq. (6.41) comes from the fact that $T^{1/(2\lceil \log T \rceil)} = e^{(\log T/(2\lceil \log T \rceil))} \leqslant \sqrt{e}$.[8]

□

[8] One can sharpen the $\log T$ term to $(\log T)^{1/2}$ in many cases, e.g., when $\alpha < 1$ and T is large enough (with respect to α). Again, we mainly focus on the polynomial terms here.

6.8.3.2 Proof of Theorem 6.9

We first provide a martingale (difference) concentration result from Wainwright (2019) (a rewrite of Theorem 2.19).

Lemma 6.17. *Let $\{D_t\}_{t=1}^{\infty}$ be a martingale difference sequence adapted to filtration $\{\mathcal{F}_t\}_{t=1}^{\infty}$. If $\mathbb{E}[\exp(\lambda D_t)|\mathcal{F}_{t-1}] \leqslant \exp(\lambda^2\sigma^2/2)$ almost surely for any $\lambda \in \mathbb{R}$, we then have*

$$\mathbb{P}\left(\left|\sum_{i=1}^{t} D_i\right| \geqslant \varepsilon\right) \leqslant 2\exp\left(-\frac{\varepsilon^2}{2t\sigma^2}\right).$$

Theorem 6.9. *For any $\alpha \in [0,1]$ unknown to the learner, run Parallel with time horizon T and optimal expected reward $\mu_\star$ leads to the following regret upper bound:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant C\ (\log T)^2\, T^{(1+\alpha)/2},$$

where C is a universal constant.

Proof. This proof largely follows the proof of Theorem 4 in Locatelli and Carpentier (2018). For any $T \in \mathbb{N}_+$ and $i \in [\lceil \log T \rceil]$, recall SR_i is the subroutine initialized with T and $\alpha_i = i/\lceil \log T \rceil$. We use $T_{i,t}$ to denote the number of samples allocated to SR_i up to time t, and represent its empirical regret at time t as $\widehat{R}_{i,t} = T_{i,t} \cdot \mu_\star - \sum_{t=1}^{T_{i,t}} X_{i,t}$ where $X_{i,t} \sim \nu_{A_{i,t}}$ is the t-th empirical reward obtained *by* SR_i and $A_{i,t}$ is the index of the t-th arm pulled *by* SR_i. We consider the corresponding regret $R_{i,t} = T_{i,t} \cdot \mu_\star - \sum_{t=1}^{T_{i,t}} \mathbb{E}[\mu_{A_{i,t}}]$ (which is random in $T_{i,t}$). We choose $\delta = 1/\sqrt{T}$ as the confidence parameter and provide $\delta' = \delta/\lceil \log T \rceil$ failure probability to each subroutine.

Notice that $R_{i,t} - \widehat{R}_{i,t} = \sum_{t=1}^{T_{i,t}} \left(X_{i,t} - \mathbb{E}[\mu_{A_{i,t}}]\right)$ is a martingale with respect to filtration $\mathcal{F}_t = \sigma\big(\bigcup_{i \in [\lceil \log T \rceil]} \{T_{i,1}, A_{i,1}, X_{i,1}, \ldots, T_{i,t}, A_{i,T_{i,t}}, X_{i,T_{i,t}}\}\big)$; and $(R_{i,t} - \widehat{R}_{i,t}) - (R_{i,t-1} - \widehat{R}_{i,t-1})$ defines a martingale difference sequence. Since, no matter what value $T_{i,t}$ takes, $X_{i,T_{i,t}} - \mathbb{E}[\mu_{A_{i,T_{i,t}}}] = (X_{i,T_{i,t}} - \mu_{A_{i,T_{i,t}}}) + (\mu_{A_{i,T_{i,t}}} - \mathbb{E}[\mu_{A_{i,T_{i,t}}}])$ is $(\sqrt{2}/2)$-sub-Gaussian (following a similar analysis as in Eq. (6.6)), applying Lemma 6.17

together with a union bound gives:

$$\mathbb{P}\left(\forall i \in [\lceil \log T \rceil], \forall t \in [T] : |\widehat{R}_{i,t} - R_{i,t}| \geqslant \sqrt{T_{i,t} \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}\right) \leqslant \delta. \quad (6.42)$$

We use $\mathcal{E} = \left\{\forall i \in [\lceil \log T \rceil], \forall t \in [T] : |\widehat{R}_{i,t} - R_{i,t}| < \sqrt{T_{i,t} \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}\right\}$ to denote the good event that holds true with probability at least $1 - \delta$. Since the regret could be trivially upper bounded by $T \cdot \delta = \sqrt{T}$ when $\mathcal{E}$ doesn't hold, we only focus on the case when event $\mathcal{E}$ holds in the following.

Fix any subroutine $k \in [\lceil \log T \rceil]$ and consider its empirical regret $\widehat{R}_{k,T}$ up to time T. For any $j \neq k$, let $T_j \leqslant T$ be the last time that the subroutine SR_j was invoked, we have

$$\begin{aligned}
\widehat{R}_{j,T_j} &\leqslant \widehat{R}_{k,T_j} \\
&\leqslant R_{k,T_j} + \sqrt{T_{k,T_j} \cdot \log\left(2T\lceil \log T \rceil/\delta\right)} \\
&\leqslant R_{k,T} + \sqrt{T \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}, \qquad (6.43)
\end{aligned}$$

where Eq. (6.43) comes from the fact that the cumulative regret $R_{k,t}$ in non-decreasing in t. Since SR_j will only run additional $\lceil \sqrt{T} \rceil$ rounds after it was selected at time T_j, we further have

$$\begin{aligned}
\widehat{R}_{j,T} &\leqslant \widehat{R}_{j,T_j} + \left\lceil \sqrt{T} \right\rceil \\
&\leqslant R_{k,T} + \sqrt{5T \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}, \qquad (6.44)
\end{aligned}$$

where Eq. (6.44) comes from the combining Eq. (6.43) with a trivial bounding $\lceil \sqrt{T} \rceil \leqslant \sqrt{4T}$ for all $T \in \mathbb{N}_+$. Combining Eq. (6.44) with the fact that $R_{j,T} \leqslant \widehat{R}_{j,T} + \sqrt{T \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}$ leads to

$$R_{j,T} \leqslant R_{k,T} + 4\sqrt{T \cdot \log\left(2T\lceil \log T \rceil/\delta\right)}. \qquad (6.45)$$

Let $i_\star \in [\lceil \log T \rceil]$ denote the index such that $\alpha_{i_\star - 1} < \alpha \leqslant \alpha_{i_\star}$. As the total regret

is the sum of all subroutines, we have that, for some universal constant C,

$$\begin{aligned}\sum_{i=1}^{\lceil \log T \rceil} R_{i,T} &\leqslant \lceil \log T \rceil \cdot \left(R_{i_\star,T} + 4\sqrt{T \cdot \log\left(2T\lceil \log T \rceil/\delta\right)} \right) && (6.46)\\ &\leqslant \lceil \log T \rceil \cdot \left(59\sqrt{e}\, \log T \cdot T^{(1+\alpha)/2} + 4\sqrt{T \cdot \log\left(2T^{3/2}\lceil \log T \rceil\right)} \right) && (6.47)\\ &\leqslant C\, (\log T)^2\, T^{(1+\alpha)/2},\end{aligned}$$

where Eq. (6.46) comes from setting $k = i_\star$ in Eq. (6.45); Eq. (6.47) comes from applying Lemma 6.8 with the non-decreasing nature of cumulative regret and taking $\delta = 1/\sqrt{T}$. Integrate once more leads to the desired result. □

6.8.3.3 Anytime Version

The anytime version of Algorithm 19 could be constructed as following.

Algorithm 21 Anytime version of Parallel

1: **for** $i = 0, 1, \ldots$ **do**
2: Run Algorithm 19 with the optimal expected reward $\mu_\star$ for 2^i rounds.

Corollary 6.18. *For any time horizon T and $\alpha \in [0, 1]$ unknown to the learner, run Algorithm 21 with optimal expected reward $\mu_\star$ leads to the following anytime regret upper:*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant C\, (\log T)^2\, T^{(1+\alpha)/2},$$

where C is a universal constant.

Proof. The proof is similar to the one for Corollary 6.11. □

7 MODEL SELECTION FOR LINEAR BANDITS

We study model selection in linear bandits, where the learner must adapt to the dimension (denoted by $d_\star$) of the smallest hypothesis class containing the true linear model while balancing exploration and exploitation. Previous papers provide various guarantees for this model selection problem, but have limitations; i.e., the analysis requires favorable conditions that allow for inexpensive statistical testing to locate the right hypothesis class or are based on the idea of "corralling" multiple base algorithms, which often performs relatively poorly in practice. These works also mainly focus on upper bounds. In this chapter, we establish the first lower bound for the model selection problem. Our lower bound implies that, even with a fixed action set, adaptation to the unknown dimension $d_\star$ comes at a cost: There is no algorithm that can achieve the regret bound $\widetilde{O}(\sqrt{d_\star T})$ simultaneously for all values of $d_\star$. We propose Pareto optimal algorithms that match the lower bound. Empirical evaluations show that our algorithm enjoys superior performance compared to existing ones.

7.1 Introduction

Model selection considers the problem of choosing an appropriate hypothesis class to conduct learning, and the hope is to optimally balance two types of error: the approximation error and the estimation error. In the supervised learning setting, the learner is provided with a (usually nested) sequence of hypothesis classes $\mathcal{H}_d \subset \mathcal{H}_{d+1}$. As an example, $\mathcal{H}_d$ could be the hypothesis class consisting of polynomials of degree at most d. The goal is to design a learning algorithm that adaptively selects the best of these hypothesis classes, denoted by $\mathcal{H}_\star$, to optimize the trade-off between approximation error and estimation error. Structural Risk Minimization (SRM) (Vapnik and Chervonenkis, 1974; Vapnik, 1995; Shawe-Taylor et al., 1998) provides a principled way to conduct model selection in the standard supervised learning setting. SRM can automatically adapt to the complexity of the

hypothesis class $\mathcal{H}_\star$, with only additional logarithmic factors in sample complexity. Meanwhile, cross-validation (Stone, 1978; Craven and Wahba, 1978; Shao, 1993) serves as a helpful tool to conduct model selection in practice.

Despite the importance and popularity of model selection in the supervised learning setting, only very recently have researchers started to study on model selection problems in interactive/sequential learning setting with bandit feedback. Two additional difficulties are highlighted in such bandit setting (Foster et al., 2019): (1) decisions/actions must be made online/sequentially without seeing the entire dataset; and (2) the learner's actions influence what data is observed, i.e., we only have partial/bandit feedback. In the simpler online learning setting with full information feedback, model selection results analogous to those in the supervised learning setting are obtained by several parameter-free online learning algorithms (McMahan and Abernethy, 2013; Orabona, 2014; Koolen and Van Erven, 2015; Luo and Schapire, 2015; Orabona and Pál, 2016; Foster et al., 2017; Cutkosky and Boahen, 2017; Cutkosky and Orabona, 2018).

The model selection problem for (contextual) linear bandits is first introduced by Foster et al. (2019). They consider a sequence of nested linear classifiers in $\mathbb{R}^{d_i}$ as the set of hypothesis classes, with $d_1 < d_2 < \cdots < d_M = d$. The goal is to adapt to the smallest hypothesis class, with apriori *unknown* dimension $d_\star$, that preserves linearity in rewards. Equivalently, one can think of the model selection problem as learning a true reward parameter $\theta_\star \in \mathbb{R}^d$, but only the first $d_\star$ entries of $\theta_\star$ contain non-zero values. The goal is to design algorithms that could automatically adapt to the intrinsic dimension $d_\star$, rather than suffering the ambient dimension d. In favorable scenarios when one can cheaply test linearity, Foster et al. (2019) provide an algorithm with regret guarantee that scales as $\widetilde{O}(K^{1/4}T^{3/4}/\gamma^2 + \sqrt{Kd_\star T}/\gamma^4)$, where K is the number of arms and γ is the smallest eigenvalue of the expected design matrix. The core idea therein is to conduct a sequential test, with sublinear sample complexity, to determine whether to step into a larger hypothesis class on the fly. Although this provides the first guarantee for model selection in the linear bandits, the regret bound is proportional to the number of arms K and the reciprocal of the smallest eigenvalue, i.e., γ^{-1}. Both K and γ^{-1} can be quite large in practice,

thus limiting the application of their algorithm. Recall that, when provided with the optimal hypothesis class, the classical algorithm LinUCB (Chu et al., 2011; Auer, 2002) for linear bandit achieves a regret bound $\widetilde{O}(\sqrt{d_\star T})$, with only polylogarithmic dependence on K and no dependence on γ^{-1}.

The model selection problem in linear bandits was further studied in many subsequent papers. We roughly divide these methods into the following two subcategories:

1. **Testing in Favorable Scenarios.** The algorithm in Ghosh et al. (2020) conducts a sequence of statistical tests to gradually estimate the true support (non-zero entries) of $\theta_\star$, and then applies standard linear bandit algorithms on identified support. The regret bound of their algorithm scales as $\widetilde{O}(d^2/\gamma^{4.65} + d_\star^{1/2} T^{1/2})$, where $\gamma = \min\{|\theta_{\star,i}| : \theta_{\star,i} \neq 0\}$ is the minimum magnitude of non-zero entries in $\theta_\star$. Their regret bound not only depends on the ambient dimension d but also scales inversely proportional to a small quantity γ. Their guarantee becomes vacuous when d and/or γ^{-1} are large. Chatterji et al. (2020) consider a different model selection problem where the rewards come from either a linear model or a model with K independent arms. Their algorithm also relies on sequential statistical testing, which requires assumptions stronger than the ones used in Foster et al. (2019) (thus suffering from similar problems).

2. **Corralling Multiple Base Algorithms.** Another approach maintains multiple base learners and use a master algorithm to determine sample allocation among base learners. This type of algorithm is initiated by the CORRAL algorithm (Agarwal et al., 2017). Focusing on our model selection setting, the base learners are usually constructed using standard linear bandit algorithms with respect to different hypothesis classes (dimensions). To give an example of the CORRAL-type of algorithm, the Smooth Corral algorithm developed in Pacchiano et al. (2020b) enjoys regret guarantees $\widetilde{O}(d_\star \sqrt{T})$ or $\widetilde{O}(d_\star^{1/2} T^{2/3})$. Other algorithms of this type, including some concurrent works, can be found

in Abbasi-Yadkori et al. (2020); Arora et al. (2020); Pacchiano et al. (2020a); Cutkosky et al. (2020, 2021).

Note that above algorithms either only work in favorable scenarios when some critical parameters, e.g., γ^{-1} and K, are not too large or must balance over multiple base algorithms which often hurts the empirical performance. They also mainly focus on developing upper bounds for the model selection problem in linear bandits. In this chapter, we explore the fundamental limits (lower bounds) of the model selection problem and design algorithms with matching guarantees (upper bounds). We establish a lower bound, using only a fixed action set, indicating that adaptation to the unknown intrinsic dimension $d_\star$ comes at a cost: There is no algorithm that can achieve the regret bound $\widetilde{O}(\sqrt{d_\star T})$ simultaneously for all values of $d_\star$. We also develop a Pareto optimal algorithm, with ideas fundamentally different from "testing" (Foster et al., 2019; Ghosh et al., 2020) and "corralling" (Pacchiano et al., 2020b; Agarwal et al., 2017), to bear on the model selection problem in linear bandits. Our algorithm is built upon the construction of virtual mixture-arms, which is previously studied in continuum-armed bandits (Hadiji, 2019) and K-armed bandits (Zhu and Nowak, 2020). We adapt their methods to our setting, with new techniques developed to deal with the linear structure, e.g., the construction of virtual dimensions.

7.1.1 Contribution and Organization

We briefly summarize our contributions as follows.

- We review the model selection problem in linear bandits, and additionally define a new parameter (in Section 7.2) that reflects the tension between time horizon and the intrinsic dimension. This parameter provides a convenient way to analyze high-dimensional linear bandits.
- We establish the first lower bound for the model selection problem in Section 7.3. Our lower bound indicates that the model selection problem is strictly harder than the problem with given optimal hypothesis class: There is no

algorithm that can achieve the non-adaptive $\widetilde{O}(\sqrt{d_\star T})$ regret bound simultaneously for all values of $d_\star$. We additionally characterize the exact Pareto frontier of the model selection problem.

- In Section 7.4, we develop a Pareto optimal algorithm that is fundamentally different from existing ones relying on "testing" or "corralling". Our algorithm is built on the construction of virtual mixture-arms and virtual dimensions. Although our main algorithm is analyzed under a mild assumption, we also provide a workaround.
- We conduct experiments in Section 7.5 to evaluate our algorithms. Our main algorithm shows superior performance compared to existing ones. We also show that our main algorithm is fairly robust to the existence of the assumption used in our analysis.

7.1.2 Additional Related Work

Bandit with large/continuous action spaces. Adaptivity issues naturally arises in bandit problems with large or infinite action space. In continuum-armed bandit problems (Agrawal, 1995), actions are embedded into a bounded subset $\mathcal{X} \subseteq \mathbb{R}^d$ with a smooth function f governing the mean payoff for each arm. Achievable theoretical guarantees are usually influenced by some smoothness parameters, and an important question is to design algorithms that adapt to these *unknown* parameters, as discussed in Bubeck et al. (2011b). Locatelli and Carpentier (2018) show that, however, no strategy can be optimal simultaneously over all smoothness classes. Hadiji (2019) establishes the Pareto frontier for continuum-armed bandits with Hölder reward functions. Adaptivity is also studied in the discrete case with a large action space (Wang et al., 2008; Lattimore, 2015; Chaudhuri and Kalyanakrishnan, 2018; Russo and Van Roy, 2018; Zhu and Nowak, 2020). Lattimore (2015) studies the Pareto frontier in standard K-armed bandits. Zhu and Nowak (2020) develop Pareto optimal algorithms for the case with multiple best arms.

High-dimensional linear bandits. As more and more complex data are being used and analyzed, modern applications of linear bandit algorithms usually involve dealing with ultra-high-dimensional data, sometimes with dimension even larger than time horizon (Deshpande and Montanari, 2012). To make progress in this high-dimensional regime, one natural idea is to study (or assume) sparsity in the reward vector and try to adapt to the unknown true support (non-zero entries). The sparse bandit problem is strictly harder than the model selection setting considered here due to the absence of the hierarchical structures. Consequently, a lower bound on the regret of the form $\Omega(\sqrt{dT})$, which scales with the ambient dimension d, is indeed unavoidable in the sparse linear bandit problem (Abbasi-Yadkori et al., 2012; Lattimore and Szepesvári, 2020). Other papers deal with the sparsity setting with additional feature feedback (Oswal et al., 2020) or further distributional/structual assumptions (Carpentier and Munos, 2012; Hao et al., 2020) to circumvent the lower bound. These high-dimensional linear bandit problems motivate our investigation of the relationship between time horizon and data dimension.

7.2 Problem Setting

We consider a linear bandit problem with a finite action set $\mathcal{A} \subseteq \mathbb{R}^d$ where $|\mathcal{A}| = K$ (Auer, 2002; Chu et al., 2011). (The feature representation of) Each arm/action $a \in \mathcal{A}$ is viewed as a d dimensional vector, and its expected reward $f(a)$ is linear with respect to a reward parameter $\theta_\star \in \mathbb{R}^d$, i.e., $f(a) = \langle a, \theta_\star \rangle$. As standard in the literature (Lattimore and Szepesvári, 2020), we assume $\max_{a \in \mathcal{A}} \|a\| \leqslant 1$ and $\|\theta_\star\| \leqslant 1$. The bandit instance is said to have intrinsic dimension $d_\star$ if $\theta_\star$ only has non-zero entries on its first $d_\star \leqslant d$ coordinates. The model selection problem aims at designing algorithm that can automatically adapt to the *unknown* intrinsic dimension $d_\star$ in the interactive learning setting with bandit feedback.

At each time step $t \in [T]$, the algorithm selects an action $A_t \in \mathcal{A}$ based on previous observations and receives a reward $X_t = \langle A_t, \theta_\star \rangle + \eta_t$, where η_t is an independent 1-sub-Gaussian noise. We define the pseudo regret (which is random, due to randomness in A_t) over time horizon T as $\widehat{R}_T = \sum_{t=1}^{T} \langle \theta_\star, a_\star - A_t \rangle$, where

$a_\star$ corresponds to the best action in action set, i.e., $a_\star = \arg\max_{a \in \mathcal{A}} \langle a, \theta_\star \rangle$. We measure the performance of any algorithm by its expected regret $R_T = \mathbb{E}\left[\widehat{R}_T\right] = \mathbb{E}\left[\sum_{t=1}^{T} \langle \theta_\star, a_\star - A_t \rangle\right]$.

We primarily focus on the high-dimensional linear bandit setting with ambient dimension d close to or even larger than (the allowed) time horizon T. We use $\mathcal{R}(T, d_\star)$ to denote the set of regret minimization problems with time horizon T and any bandit instance with intrinsic dimension $d_\star$. We emphasize that T is part of the problem instance, which was largely neglected in previous work focusing on the low dimensional regime where $T \gg d_\star$. To model the tension between the allowed time horizon and the intrinsic dimension, we define the hardness level as

$$\psi\left(\mathcal{R}(T, d_\star)\right) = \min\{\alpha \geqslant 0 : d_\star \leqslant T^\alpha\} = \log d_\star / \log T.$$

$\psi(\mathcal{R}(T, d_\star))$ is used here since it precisely captures the regret over the set of regret minimization problem $\mathcal{R}(T, d_\star)$, as discussed later in our review of the LinUCB algorithm and the lower bound. Since smaller $\psi(\mathcal{R}(T, d_\star))$ indicates easier problem, we define the family of regret minimization problems with *hardness level* at most α as

$$\mathcal{H}_T(\alpha) = \{\cup \mathcal{R}(T, d_\star) : \psi(\mathcal{R}(T, d_\star)) \leqslant \alpha\},$$

where $\alpha \in [0, 1]$. Although T is necessary to define a regret minimization problem, the hardness of the problem is encoded into a single parameter α: Problems with different time horizons but the same α are equally difficult in terms of the regret achieved by LinUCB (the exponent of T). We explore the connection $d_\star \leqslant T^\alpha$ in the rest of this chapter and focus on (polynomial) dependence on T (i.e., the dependence on $d_\star$ is translated into the dependence on T^α). We are interested in designing algorithms with worst case guarantees over $\mathcal{H}_T(\alpha)$, but *without* the knowledge of α.

LinUCB and upper bounds. In the standard setting where $d_\star$ is known, LinUCB Chu et al. (2011); Auer (2002) achieves $\widetilde{O}(\sqrt{d_\star T})$ regret.[1] For any problem in $\mathcal{H}_T(\alpha)$ with *known* α, one could run LinUCB on the first $\lfloor T^\alpha \rfloor$ coordinates and achieve $\widetilde{O}(T^{(1+\alpha)/2})$ regret. The goal of model selection is to achieve the $\widetilde{O}(T^{(1+\alpha)/2})$ regret but without the knowledge of α.

Lower bounds. In the case when $d_\star \leqslant \sqrt{T}$, Chu et al. (2011) prove a $\Omega(\sqrt{d_\star T})$ lower bound for linear bandits. When $d_\star \geqslant \sqrt{T}$ is the case, a lower bound $\Omega(K^{1/4}T^{3/4})$ is developed in Abe et al. (2003).

7.3 Lower Bound and Pareto Optimality

We study lower bounds for model selection in this section. We show that simultaneously adapting to all hardness levels is impossible. Such fundamental limitation leads to the established of Pareto frontier.

Our lower bound is constructed by relating the regrets between two (sets of) closely related problems: We show that any algorithm achieves good performance on one of them necessarily performs bad on the other one. Similar ideas are previously explored in continuum-armed bandit and K-armed bandits (Locatelli and Carpentier, 2018; Hadiji, 2019; Zhu and Nowak, 2020). We study the linear case with model selection and establish the following lower bound.[2] We use $\omega \in \mathcal{H}_T(\alpha)$ to represent any bandit regret minimization problem with time horizon T and hardness level at most α (i.e., $d_\star \leqslant T^\alpha$).

Theorem 7.1. *Consider any* $0 \leqslant \alpha' < \alpha \leqslant 1$ *and* $B > 0$ *satisfying* $T^\alpha \leqslant B$ *and* $\lfloor T^\alpha/2 \rfloor \geqslant \max\{T^\alpha/4, T^{\alpha'}, 2\}$. *If an algorithm is such that* $\sup_{\omega \in \mathcal{H}_T(\alpha')} R_T \leqslant B$, *then the*

[1] Technically, the regret bound is only achieved by a more complicated algorithm SupLinUCB. However, it's common to use LinUCB as the practical algorithm. See Chu et al. (2011) for detailed discussion.

[2] Our lower bound is quantitatively similar to the one studied in K-armed bandits with multiple best arms (Zhu and Nowak, 2020).

regret of the same algorithm must satisfy

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \geqslant 2^{-10} T^{1+\alpha} B^{-1}. \tag{7.1}$$

Our lower bound delivers important messages to the model selection problem in linear bandits. Most of the previous efforts and open problems (Foster et al., 2019; Pacchiano et al., 2020b) are made to match the usual non-adaptive regret with known $d_\star$ (or α). Our lower bound, however, provides a negative answer towards the open problem of achieving regret guarantees $\widetilde{O}(T^{(1+\alpha)/2})$ simultaneously for all hardness levels α. We interpret this result next.

Interpretation of Theorem 7.1. Fix any linear bandit algorithm. We consider two problem instances with different hardness levels $0 \leqslant \alpha' < \alpha \leqslant 1$ (and satisfy the constrains in Theorem 7.1). On one hand, if the algorithm is such that $\sup_{\omega \in \mathcal{H}_T(\alpha')} R_T = \widetilde{\omega}(T^{(1+\alpha')/2})$, we know that this algorithm is already sub-optimal over problems with hardness level at most α'. On the other hand, suppose that the algorithm achieves the desired regret $\widetilde{O}(T^{(1+\alpha')/2})$ over $\mathcal{H}_T(\alpha')$. Eq. (7.1) then tells us that $\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T = \widetilde{\Omega}(T^{(1+2\alpha-\alpha')/2})$, which is (asymptotically) larger than the desired regret $\widetilde{O}(T^{(1+\alpha)/2})$ over problems with hardness level at most α.

If we aim at providing regret bounds with only polylogarithmic dependence on K in linear bandits (which is usually the case for linear bandits with finite action set (Auer, 2002; Chu et al., 2011)). our lower bound also provides a negative answer to the open problem of achieving a weaker guarantee $\widetilde{O}(T^\gamma d_\star^{1-\gamma}) = \widetilde{O}(T^{\gamma+\alpha(1-\gamma)})$, with $\gamma \in [1/2, 1)$ (Foster et al., 2019), simultaneously for all $d_\star$ (or α).

In the model selection setting, the performance of any algorithm should be a function of the hardness level α: The algorithm needs to adapt the *unknown* α. To further explore the fundamental limit for model selection in linear bandits, following Hadiji (2019); Zhu and Nowak (2020), we define rate function to capture the performance of any algorithm (in terms of its regret dependence on polynomial terms of T).

Definition 7.2. *Let* $\theta : [0,1] \to [0,1]$ *denote a non-decreasing function. An algorithm achieves the rate function* θ *if*

$$\forall \varepsilon > 0, \forall \alpha \in [0,1], \quad \limsup_{T \to \infty} \frac{\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T}{T^{\theta(\alpha)+\varepsilon}} < +\infty.$$

Since there may not always exist a pointwise ordering over rate functions, we consider the notion of Pareto optimality over rate functions.

Definition 7.3. *A rate function* θ *is Pareto optimal if it is achieved by an algorithm, and there is no other algorithm achieving a strictly smaller rate function* θ' *in the pointwise order. An algorithm is Pareto optimal if it achieves a Pareto optimal rate function.*

We establish the following lower bound for any rate function that can be achieved by an algorithm designed for model selection in linear bandits.

Theorem 7.4. *Suppose a rate function* θ *is achieved by an algorithm, then we must have*

$$\theta(\alpha) \geqslant \min\{\max\{\theta(0), 1 + \alpha - \theta(0)\}, 1\}, \tag{7.2}$$

with $\theta(0) \in [1/2, 1]$.

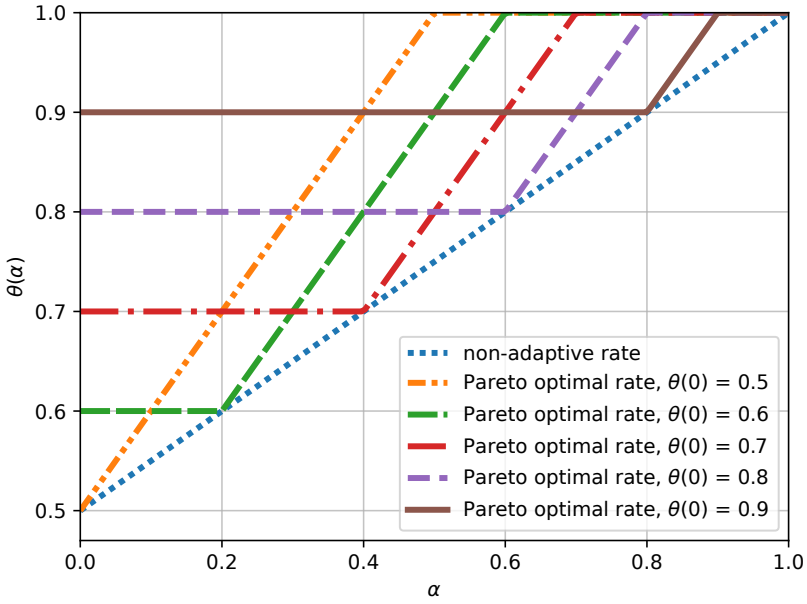


Figure 7.1: Pareto optimal rates for model selection in linear bandits.

Fig. 7.1 illustrates the Pareto frontiers for the model selection problem in linear bandits: The blue dashed line represents the non-adaptive rate function achieved by LinUCB with *known* α; Other curves represent Pareto optimal rate functions (achieved by Pareto optimal algorithms introduced in Section 7.4) for the model selection problem in linear bandits. Fig. 7.1 implies that no algorithm can achieve the non-adaptive rate simultaneously for all α: any Pareto optimal curve has to be higher than the non-adaptive curve at least at some points.

Pareto optimality of CORRAL-type of algorithms. We remark that, accompanied with our lower bound, the Smooth Corral algorithm presented in Pacchiano et al. (2020b) is also Pareto optimal. While only a $\widetilde{O}(d_\star\sqrt{T})$ regret bound is presented for the Smooth Corral algorithm, upon inspection of their analysis, we find that Smooth Corral can actually match the lower bound in Eq. (7.2) by setting the learning rate as $\eta = T^{-\theta(0)}$, for any $\theta(0) \in [1/2, 1)$. See Section 7.7.3.3 for a detailed

discussion.

Although the CORRAL-type of algorithm (e.g., Smooth Corral) is Pareto optimal, they may not be effective in problems with specific structures (Papini et al., 2021). We introduce a new Pareto optimal algorithm in the next section, which is shown to be more practical than Smooth Corral regarding model selection problems in linear bandits (see Section 7.5).

7.4 Pareto Optimality with New Ideas

We develop a Pareto optimal algorithm LinUCB++ (Algorithm 22) that operates fundamentally different from algorithms rely on "testing" (Foster et al., 2019; Ghosh et al., 2020) or "corralling" (Pacchiano et al., 2020b; Agarwal et al., 2017). Our algorithm is built upon the construction of virtual mixture-arms (Hadiji, 2019; Zhu and Nowak, 2020) and virtual dimensions.

We first introduce some additional notations. For any vector $a \in \mathbb{R}^d$ and $0 \leqslant d_i \leqslant d$, we use $a^{(d_i)} \in \mathbb{R}^{d_i}$ to represent the truncated version of a that only keeps the first d_i dimensions. We also use $[a_1; a_2]$ to represent the concatenated vector of a_1 and a_2. We denote $\mathcal{A}^{(d_i)} \subseteq \mathbb{R}^{d_i}$ as the "truncated" action (multi-) set, i.e., $\mathcal{A}^{(d_i)} = \left\{a^{(d_i)} \in \mathbb{R}^{d_i} : a \in \mathcal{A}\right\}$. One can always *manually* construct the truncated action set $\mathcal{A}^{(d_i)}$ and *pretend* to work with arms with truncated feature representations (though their expected rewards may not be aligned with the truncated feature representations).

We present LinUCB++ in Algorithm 22. LinUCB++ operates in iterations with geometrically increasing length, and it invokes LinUCB (SupLinUCB) (Chu et al., 2011; Auer, 2002) with (roughly) geometrically decreasing dimensions. The core steps of LinUCB++ are summarized at lines 3 and 4 in Algorithm 22, which consists of construction of virtual mixture-arms and virtual dimensions (the modified linear bandit problem). We next explain in detail these two core ideas.

The virtual mixture-arm. After each iteration j, let $\widehat{p}_j$ denote the vector of empirical sampling frequencies of the arms in that iteration, i.e., the k-th element

Algorithm 22 LinUCB++

Input: Time horizon T and a user-specified parameter $\beta \in [1/2, 1)$.

1: **Set:** $p = \lceil \log_2 T^{\beta} \rceil$, $d_i = \min\{2^{p+2-i}, d\}$ and $\Delta T_i = \min\{2^{p+i}, T\}$.
2: **for** $i = 1, \ldots, p$ **do**
3: Run LinUCB on a set of arms S_i for ΔT_i rounds, where S_i contains all arms in $\mathcal{A}^{(d_i)}$ *and* a set of virtual mixture-arms constructed from previous iterations, i.e., $\{\widetilde{\nu}_j\}_{j<i}$. LinUCB is operated with respect to an modified linear bandit problem with added virtual dimensions.
4: Construct a virtual mixture-arm $\widetilde{\nu}_i$ based on empirical sampling frequencies in iteration i.

of $\widehat{p}_j$ is the number of times arm k, including all previously constructed virtual mixture-arms, was sampled in iteration j divided by the total number of time steps ΔT_j. The virtual mixture-arm for iteration j is the $\widehat{p}_j$-mixture of the arms played in iteration j, denoted by $\widetilde{\nu}_j$. When LinUCB samples from $\widetilde{\nu}_j$, it first draws a real arm $j_t \sim \widehat{p}_j$ with feature representation A_t,[3] then pull the real arm A_t to obtain a reward $X_t = \langle \theta_\star, A_t \rangle + \eta_t$. The expected reward of virtual mixture-arm $\widetilde{\nu}_j$ can be expressed as $\langle \theta_\star, a_\star \rangle - R_{\Delta T_j}/\Delta T_j$, where we use $R_{\Delta T_j}$ to denote the expected regret suffered in iteration j. Virtual mixture-arms $\widetilde{\nu}_j$ provide a convenient summary of the information gained in the j-th iterations so that we don't need to explore arms in the (effectively) d_j dimensional space again.

Linear bandits with added virtual dimensions. We consider the linear bandit problem in iteration i, where each arm in $\mathcal{A}^{(d_i)}$ is viewed as a vector in $\mathbb{R}^{d_i}$. Besides this simple truncation, we lift the feature representation of each arm into a slightly higher dimensional space to include the $i-1$ virtual mixture-arms constructed in previous iterations (i.e., adding virtual dimensions). More specifically, we augment $i-1$ zeros to the feature representation of each truncated real arm $a \in \mathcal{A}^{(d_i)}$; we also view each virtual mixture-arm $\widetilde{\nu}_j$ as a $d_i + i - 1$ dimensional vector $\widetilde{\nu}_j^{\langle d_i \rangle}$ with its $(d_i + j)$-th entry being 1 and all other entries being 0. As a result, LinUCB will operate on an modified linear bandit problem with action set $\mathcal{A}^{\langle d_i \rangle} \subseteq \mathbb{R}^{d_i+i-1}$,

[3] If the index of another virtual mixture-arm is returned, we sample from that virtual mixture-arm until a real arm is returned.

where $\mathcal{A}^{\langle d_i \rangle} = \left\{[a^{(d_i)}; 0] \in \mathbb{R}^{d_i+i-1} : a \in \mathcal{A}\right\} \cup \left\{\widetilde{v}_j^{\langle d_i \rangle}\right\}$, and $|\mathcal{A}^{\langle d_i \rangle}| = K + i - 1$. Working with added virtual dimensions allows us to incorporate information stored in virtual mixture-arms without too much additional cost since $i \leqslant p = O(\log T)$.

Remark 7.5. *Previous application of the virtual mixture-arms only works in continuum-armed bandits or* K*-armed bandits (Zhu and Nowak, 2020; Hadiji, 2019), where no further modifications are needed to incorporate information stored in virtual mixture-arms. Besides the construction of the virtual dimension, we also provide another way to incorporate the virtual mixture-arms in Section 7.4.2. These modifications are important for the linear bandit case.*

7.4.1 Analysis

We first analyze LinUCB++ with the following assumption. A modified version of LinUCB++ (Algorithm 23) is provided in Section 7.4.2 and analyzed without the assumption.

Assumption 7.6. *An action set $\mathcal{A} \subseteq \mathbb{R}^d$ is expressive if we have $a^{[d_i]} = [a^{(d_i)}; 0] \in \mathcal{A}$ for any $a \in \mathcal{A}$ and $d_i < d$.*

Assumption 7.6 is naturally satisfied when certain combinatorial structure and ranking information are associated with the action set. This is best explained with an example. Suppose the arms are consumer products and each has a subset of d possible features, i.e., the arms are *binary vectors* in $\mathbb{R}^d$ indicating the features of the product (the combinatorial aspect). Think of the features as being ordered from base-level features to high-end features (the ranking information). In this case, Assumption 7.6 means that if a product $a \in \mathcal{A}$, then $\mathcal{A}$ also contains all products with fewer high-end features, i.e., truncations of action a. We also make the following two comments regarding Assumption 7.6.

1. The action set we used to construct the lower bound in Theorem 7.1 can be made expressive, as noted in Remark 7.10 in Section 7.7.1.1;

2. Although the original version of LinUCB++ is analyzed with Assumption 7.6, it shows strong empirical performance even without such assumption (see Section 7.5).

Equipped with Assumption 7.6, we can replace the "truncated" action set $\mathcal{A}^{(d_i)}$ with real arms that actually exist in the action set. As a result, the linearity in rewards is preserved in the modified linear bandit problem in $\mathbb{R}^{d_i+i-1}$ with added virtual dimensions. The modified linear bandit problem is associated with reward vector $\theta_\star^{\langle d_i \rangle} = \left[\theta_\star^{(d_i)}; \widetilde{\mu}_1; \ldots; \widetilde{\mu}_{i-1}\right] \in \mathbb{R}^{d_i+i-1}$, where we use $\widetilde{\mu}_i = \langle \theta_\star, a_\star \rangle - R_{\Delta T_i}/\Delta T_i$ to denote the expected reward of mixture-arm $\widetilde{\nu}_i$. In the i-th iteration of LinUCB++, we invoke LinUCB to learn reward vector $\theta_\star^{\langle d_i \rangle} \in \mathbb{R}^{d_i+i-1}$, which takes worst case regret proportional to $d_i + i - 1$ instead of the ambient dimension d.

Since there are at most $O(\log T)$ iterations of LinUCB++, we only need to upper bound its regret at each iteration. Suppose S_i is the set of actions that LinUCB++ is working on at iteration i. We use $a_{S_i} = \arg\max_{a \in S_i} \langle \theta_\star, a \rangle$ to denote the arm with the highest expected reward; and decompose the regret into approximation error and learning error:

$$R_{\Delta T_i} = \underbrace{\mathbb{E}\left[\Delta T_i \cdot \langle \theta_\star, a_\star - a_{S_i} \rangle\right]}_{\text{expected approximation error due to the selection of } S_i} + \underbrace{\mathbb{E}\left[\sum_{t=1}^{\Delta T_i} \langle \theta_\star, a_{S_i} - A_t \rangle\right]}_{\text{expected learning error due to the sampling rule } \{A_t\}_{t=1}^T} . \tag{7.3}$$

The learning error. At each iteration i, LinUCB++ invokes LinUCB on a linear bandit problem in $\mathbb{R}^{d_i+i-1}$ for ΔT_i time steps, where d_i and ΔT_i are specifically chosen such that $d_i \Delta T_i \leqslant \widetilde{O}(T^{2\beta})$. The learning error is then upper bounded by $\widetilde{O}(\sqrt{d_i \Delta T_i}) = \widetilde{O}(T^\beta)$ based on the regret bound of LinUCB (the norm of reward vector $\theta_\star^{\langle d_i \rangle}$ increases with iteration i due to added virtual dimensions, we deal with that in Section 7.7.2.2).

The approximation error. Let $i_\star \in [p]$ denote the largest integer such that $d_{i_\star} \geqslant d_\star$. For iterations $i \leqslant i_\star$, since $\theta_\star$ only has its first $d_\star \leqslant d_i$ coordinates being non-zero, we have $\max_{a \in \mathcal{A}^{\langle d_i \rangle}} \{\left\langle \theta_\star^{\langle d_i \rangle}, a \right\rangle\} = \langle \theta_\star, a_\star \rangle$ and the expected approximation error equals zero. As a result, we upper bound the expected regret for iteration $i \leqslant i_\star$ by its expected learning error, i.e., $R_{\Delta T_i} \leqslant \widetilde{O}(T^\beta)$. Now consider any iteration $i > i_\star$. Since the virtual mixture-arm $\widetilde{\nu}_{i_\star}$ is constructed by then, and its expected reward is $\widetilde{\mu}_{i_\star} = \langle \theta_\star, a_\star \rangle - R_{\Delta T_{i_\star}}/\Delta T_{i_\star}$, we can further bound the expected approximation error by $\Delta T_i R_{\Delta T_{i_\star}}/\Delta T_{i_\star} = \widetilde{O}(T^{1+\alpha-\beta})$ (detailed in Section 7.7.2.5).

We now present the formal guarantees of LinUCB++.

Theorem 7.7. *Run LinUCB++ with time horizon T and any user-specified parameter $\beta \in [1/2, 1)$ leads to the following upper bound on the expected regret:*

$$\begin{aligned} &\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \\ &= O\left(\log^{7/2}(KT \log T) \cdot T^{\min\{\max\{\beta, 1+\alpha-\beta\}, 1\}}\right). \end{aligned}$$

The next theorem shows that LinUCB++ is Pareto optimal with *any* input $\beta \in [1/2, 1)$.

Theorem 7.8. *The rate function achieved by LinUCB++ with any input $\beta \in [1/2, 1)$, i.e.,*

$$\theta_\beta : \alpha \mapsto \min\{\max\{\beta, 1 + \alpha - \beta\}, 1\}, \tag{7.4}$$

is Pareto optimal.

7.4.2 Removing Assumption 7.6

Assumption 7.6 is used to preserve linearity when working with truncated action sets. In general, one should not expect to deal with misspecified linear bandits without extra cost: Lattimore et al. (2020) develop a regret lower bound $\Omega(\varepsilon\sqrt{d}\,T)$ for misspecified linear bandits with misspecification level ε. The lower bound scales linearly with T if there is no extra control/assumptions on the misspecified level ε.

Going back to our algorithm, however, we notice that there is a special structure in the source of misspecifications: the virtual-mixture arms are never misspecified. We explore this fact and provide a modified version of Algorithm 22 (i.e., Algorithm 23) that works *without* Assumption 7.6 and is Pareto optimal. The modified algorithm is less practical since it invokes Smooth Corral as a subroutine (see Section 7.5).

Algorithm 23 LinUCB++ with Corral

Input: Time horizon T and a user-specified parameter $\beta \in [1/2, 1)$.

1: **Set:** $p = \lceil \log_2 T^{\beta} \rceil$, $d_i = \min\{2^{p+2-i}, d\}$ and $\Delta T_i = \min\{2^{p+i}, T\}$.
2: **for** $i = 1, \ldots, p$ **do**
3: Construct two (smoothed) base algorithms: (1) a LinUCB algorithm working with action set $\mathcal{A}^{(d_i)}$; and (2) a UCB algorithm working with the set of virtual mixture-arms (if any), i.e., $\{\widetilde{\nu}_j\}_{j<i}$. Invoke Smooth Corral as the master algorithm with learning rate $\eta = 1/\sqrt{d_i \Delta T_i}$.
4: Construct a virtual mixture-arm $\widetilde{\nu}_i$ based on the empirical sampling frequencies in iteration i.

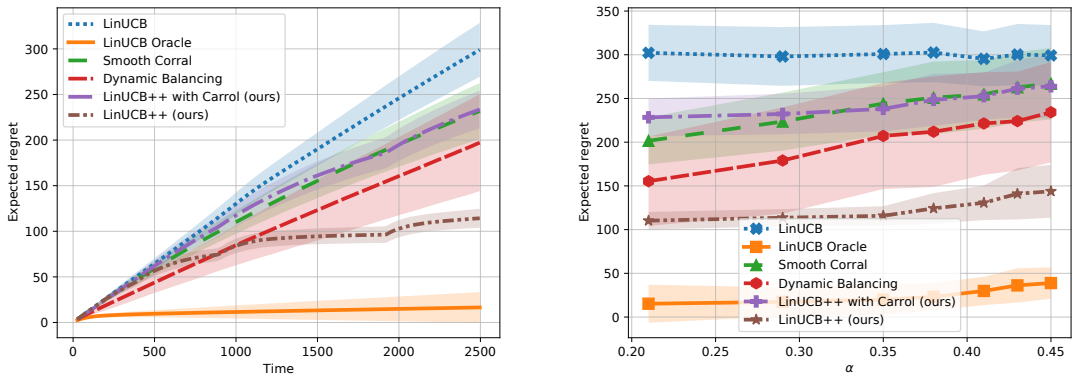
We defer detailed discussion on Algorithm 23 and Smooth Corral to Section 7.7.3. We state the guarantee of Algorithm 23 next.

Theorem 7.9. *With any input $\beta \in [1/2, 1)$, the rate function achieved by Algorithm 23 (without Assumption 7.6) is Pareto optimal.*

7.5 Empirical Results

We empirically evaluate our algorithms LinUCB++ and LinUCB++ with Corral in this section. We find that LinUCB++ enjoys superior performance compared to existing algorithms. Although Assumption 7.6 is needed in the analysis of LinUCB++, our experiments show that LinUCB++ is fairly robust to the existence of such assumption.

We compare LinUCB++ and LinUCB++ with Corral with four baselines: LinUCB (Chu et al., 2011), LinUCB Oracle, Smooth Corral (Pacchiano et al., 2020b) and Dynamic Balancing (Cutkosky et al., 2021). LinUCB is the standard linear bandit algorithm



(a) (b)

Figure 7.2: Experiments *without* Assumption 7.6. (a) Comparison of progressive regret curve with hardness level $\alpha \approx 0.32$. (b) Comparison of regret with varying α.

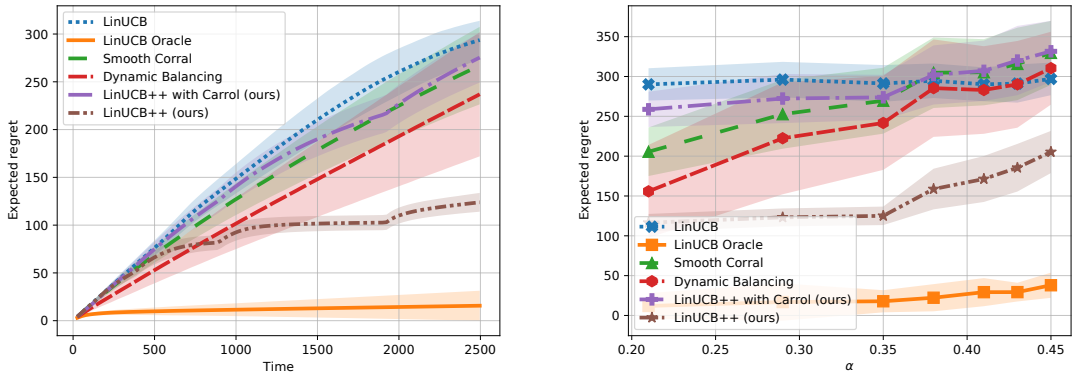
that works in the ambient dimension $\mathbb{R}^d$. LinUCB Oracle represents the oracle version of LinUCB: it takes the knowledge of the instrinsic dimension $d_\star$ and works in $\mathbb{R}^{d_\star}$. Smooth Corral and Dynamic Balancing are implemented with $M = \lceil \log_2 d \rceil$ base LinUCB learners with different dimensions $d_i \in \{2^0, 2^1, \ldots, 2^{M-1}\}$; their master algorithms conduct corraling/regret balancing on top of these base learners. We set $\beta = 0.5$ in LinUCB++ and LinUCB++ with Corral.[4] The regularization parameter λ for least squares in (all subroutines/base learners of) LinUCB is set as 0.1.

We first conduct experiments *without* an expressive action set (i.e., without Assumption 7.6). We consider a regret minimization problem with time horizon $T = 2500$ and a bandit instance consists of $K = 1200$ arms selected uniformly at random in the $d = 600$ dimensional unit ball. We set reward parameter $\theta_\star = [1/\sqrt{d_\star}, \ldots, 1/\sqrt{d_\star}, 0, \ldots, 0]^\top \in \mathbb{R}^d$ for any intrinsic dimension $d_\star$ (see Section 7.7.4 for experiments with other choices of $\theta_\star$). To prevent lengthy exploration over exploitation, we consider Gaussian noises with zero means and 0.1 standard deviations. We evaluate each algorithm on 100 independent trials and average

[4] In practice, we recommend taking $\beta = (1 + \widehat{\alpha})/2$ if an estimation $\widehat{\alpha}$ (of α) is available; otherwise, we empirically find that taking $\beta = 0.5$ works well.

the results. Fig. 7.2a shows how regret curves of different algorithms increase. The experiment is run with intrinsic dimension $d_\star = 12$, which corresponds to a hardness level $\alpha \approx 0.32$. LinUCB++ outperforms all other algorithms (except LinUCB Oracle), and enjoys the smallest variance. LinUCB++ (almost) flatten its regret curve at early stages, indicating that it has learned the true reward parameter. Fig. 7.2b illustrates the performance of algorithms with respect to different intrinsic dimensions. We run experiments with $d_\star \in \{5, 10, 15, 20, 25, 30, 35\}$, and mark the corresponding α values in the plot. Across all α values, LinUCB++ shows superior performance compared to LinUCB, Smooth Corral, Dynamic Balancing and LinUCB++ with Corral. These results indicate that LinUCB++ can be practically applied without an expressive action set (thus without Assumption 7.6).

The empirically poor performance of CORRAL-type of algorithms might be due to the fact that they need to balance over multiple base algorithms. On the other hand, LinUCB++ invokes only one LinUCB subroutine at each iteration. Although the subroutine is restarted at the beginning of each iteration, it runs on (roughly) geometrically decreasing dimensions. Such efficient learning procedure is backed by our construction of virtual mixture-arms and virtual dimensions.



(a) (b)

Figure 7.3: Similar experiment setups to those shown in Fig. 7.2, but with Assumption 7.6.

We also run experiments *with* expressive action sets. We first generate $K = 800$ arms uniformly at random from a $d = 400$ dimensional unit ball. The action set is then made *expressive* by adding actions with truncated features.[5] We provide the expressive action set to all algorithms since the best reward could be achieved by a truncated arm. Other experimental setups are similar to the ones described before. The shape of curves appearing in both Fig. 7.3a and Fig. 7.3b are resembles the ones in Fig. 7.2, and LinUCB++ outperforms LinUCB, Smooth Corral, Dynamic Balancing and LinUCB++ with Corral. One slight difference is that Smooth Corral, Dynamic Balancing, LinUCB++ with Corral and LinUCB++ have relatively worse performance when as α increases: The regret curves (in Fig. 7.3b) increase at faster speeds. Smooth Corral, Dynamic Balancing and LinUCB++ with Corral are outperformed by the standard LinUCB when the hardness level α gets large.

7.6 Discussion

We study the model selection problem in linear bandits where the goal is to adapt to the *unknown* intrinsic dimension $d_\star$, rather than suffering from regret proportional to the ambient dimension d. We establish a lower bound indicating that adaptation to the unknown intrinsic dimension $d_\star$ comes at a cost: There is no algorithm that can achieve the regret bound $\widetilde{O}(\sqrt{d_\star T})$ simultaneously for all values of $d_\star$. Under a mild assumption, we design a Pareto optimal algorithm, with ideas fundamentally different from "testing" (Foster et al., 2019; Ghosh et al., 2020) and "corralling" (Pacchiano et al., 2020b; Agarwal et al., 2017), to bear on the model selection problem in linear bandits. We also provide a workaround to remove the assumption. Experimental evaluations show superior performance of our main algorithm compared to existing ones.

Although linear bandits with a fixed action set are commonly studied in the literature (Lattimore et al., 2020; Wagenmaker et al., 2021), an interesting direction is to generalize LinUCB++ to the contextual setting. The current version of LinUCB++

[5] We only truncate actions with respect to d_i s selected by LinUCB++ to avoid the computational burden of dealing with a large number of actions.

works in the setting with adversarial contexts under the following two additional assumptions: (1) we have a nested sequence of action sets $\mathcal{A}_t \subseteq \mathcal{A}_{t+1}$ with $|\mathcal{A}_T| \leqslant K$; and (2) one of the best/near-optimal arm belongs to $\mathcal{A}_1$. How to remove/weaken these assumptions is left to future work. We also remark that, after our initial (arXiv) publication, Marinov and Zimmert (2021) established the Pareto frontier for general contextual bandits, providing a negative answer to open problems raised in Foster et al. (2020b).

7.7 Proofs and Supporting Results

7.7.1 Proofs and Supporting Results for Section 7.3

Besides specific treatments for linear bandits (e.g., the lower bound construction for model selection), our proofs for this section largely follow the ones developed in Hadiji (2019); Zhu and Nowak (2020). We provide details here for completeness.

7.7.1.1 Proof of Theorem 7.1

We consider $K+1$ linear bandit instances such that each is characterized by a reward vector $\theta_i \in \mathbb{R}^d$, $0 \leqslant i \leqslant K$, with different intrinsic dimensions $d_\star$ (or equivalently α). For any action $a \in \mathbb{R}^d$, we obtain a reward $r = \langle \theta_i, a \rangle + \eta$ where η is an independent $(1/2)$-sub-Gaussian noise. Time horizon T is fixed and the ambient dimension d is assumed to be large enough to avoid some trivial conflicts in the following construction (e.g., we need $d \geqslant T^\alpha$ to construct θ_i) . For any $0 \leqslant \alpha' < \alpha \leqslant 1$ so that $T^\alpha/2 \geqslant T^{\alpha'}$, we now provide an explicit construction of $\{\theta_i\}_{i=0}^K$ as followings, with $\Delta \in \mathbb{R}$ to be specified later.

1. Let $\theta_0 \in \mathbb{R}^d$ be any vector such that it is only supported on one of its first $\lfloor T^{\alpha'} \rfloor$ coordinates and $\|\theta_0\|_2 = \Delta/2$. The regret minimization problem with respect to θ_0 belongs to $\mathcal{H}_T(\alpha')$ by construction.

2. For any $i \in [K]$, let $\theta_i = \theta_0 + \Delta \cdot e_{\rho(i)}$ where e_j is the j-th canonical base and $\rho(i) = \lfloor T^\alpha/2 \rfloor + i$. We set $K = \lfloor T^\alpha/2 \rfloor = \Theta(T^\alpha)$ so that the regret minimization problem with respect to any θ_i belongs to $\mathcal{H}_T(\alpha)$.

We consider a common *fixed* action set $\mathcal{A} = \{a_i\}_{i=0}^K = \{\theta_0/\|\theta_0\|\} \cup \{e_{\rho(i)}\}_{i=1}^K$ for all regret minimization problems (we set $a_0 = \theta_0/\|\theta_0\|$ and $a_i = e_{\rho(i)}$ for convenience). We could notice that a_0 is the best arm with respect to θ_0, which has expected reward $\Delta/2$; and a_i is the best arm with respect to θ_i, which has expected reward Δ.

Remark 7.10. *The action set $\mathcal{A}$ can be made expressive by augmenting the action set with an all-zero action. The all-zero action will not affect our analysis since it always has zero expected reward.*

Remark 7.11. *One can also add other canonical bases into the action set $\mathcal{A}$ so that $\{\theta_i\}_{i=1}^K$ becomes the unique reward vector for corresponding problems. These additional actions will not affect our analysis as well since they all have zero expected reward.*

For any $t \in [T]$, the tuple of random variables $H_t = (A_1, X_1, \ldots, A_t, X_t)$ is the outcome of an algorithm interacting with an bandit instance up to time t. Let $\Omega_t = \prod_{i=1}^t (\mathcal{A} \times \mathbb{R})$ and $\mathcal{F}_t = \mathfrak{B}(\Omega_t)$; one could then define a measurable space $(\Omega_t, \mathcal{F}_t)$ for H_t. The random variables $A_1, X_1, \ldots, A_t, X_t$ that make up the outcome are defined by their coordinate projections:

$$A_t(a_1, x_1, \ldots, a_t, x_t) = a_t \quad \text{and} \quad X_t(a_1, x_1, \ldots, a_t, x_t) = x_t.$$

For any fixed algorithm/policy π and bandit instance θ_i, we are now constructing a probability measure $\mathbb{P}_{i,t}$ over $(\Omega_t, \mathcal{F}_t)$. Note that a policy π is a sequence $(\pi_t)_{t=1}^T$, where π_t is a probability kernel from $(\Omega_{t-1}, \mathcal{F}_{t-1})$ to $(\mathcal{A}, 2^{\mathcal{A}})$ with the first probability kernel $\pi_1(\omega, \cdot)$ being defined arbitrarily over $(\mathcal{A}, 2^{\mathcal{A}})$, to model the selection of the first action. For each i, we define another probability kernel $p_{i,t}$ from $(\Omega_{t-1} \times \mathcal{A}, \mathcal{F}_{t-1} \otimes 2^{\mathcal{A}})$ to $(\mathbb{R}, \mathfrak{B}(\mathbb{R}))$ that models the reward. Since the reward is distributed according to $\mathcal{N}(\theta_i^\top a_t, 1/4)$, we gives its explicit expression for any $B \in \mathfrak{B}(\mathbb{R})$ as

following

$$p_{i,t}\big((a_1, x_1, \ldots, a_t), B\big) = \int_B \sqrt{\frac{2}{\pi}} \exp\big(-2(x - \theta_i^\top a_t)\big) dx.$$

The probability measure over $\mathbb{P}_{i,t}$ over $(\Omega_t, \mathcal{F}_t)$ could then be define recursively as $\mathbb{P}_{i,t} = p_{i,t}\big(\pi_t \mathbb{P}_{i,t-1}\big)$. We use $\mathbb{E}_i$ to denote the expectation taken with respect to $\mathbb{P}_{i,T}$. We have the following lemmas.

Lemma 7.12 (Lattimore and Szepesvári (2020))**.**

$$\mathrm{KL}\left(\mathbb{P}_{0,T}, \mathbb{P}_{i,T}\right) = \mathbb{E}_0\left[\sum_{t=1}^{T} \mathrm{KL}\left(\mathcal{N}(\theta_0^\top A_t, 1/4), \mathcal{N}\left(\theta_i^\top A_t, 1/4\right)\right)\right]. \tag{7.5}$$

Lemma 7.13 (Hadiji (2019))**.** *Let $\mathbb{P}$ and $\mathbb{Q}$ be two probability measures. For any random variable $Z \in [0,1]$, we have*

$$|\mathbb{E}_{\mathbb{P}}[Z] - \mathbb{E}_{\mathbb{Q}}[Z]| \leqslant \sqrt{\frac{\mathrm{KL}(\mathbb{P}, \mathbb{Q})}{2}}.$$

Theorem 7.1. *Consider any $0 \leqslant \alpha' < \alpha \leqslant 1$ and $B > 0$ satisfying $T^\alpha \leqslant B$ and $\lfloor T^\alpha/2 \rfloor \geqslant \max\{T^\alpha/4, T^{\alpha'}, 2\}$. If an algorithm is such that $\sup_{\omega \in \mathcal{H}_T(\alpha')} R_T \leqslant B$, then the regret of the same algorithm must satisfy*

$$\sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \geqslant 2^{-10} T^{1+\alpha} B^{-1}. \tag{7.1}$$

Proof. Let $N_i(T) = \sum_{t=1}^{T} \mathbb{1}\left(A_t = a_i\right)$ denote the number of times the algorithm π selects arm a_i up to time T. Let $R_{i,T}$ define the expected regret achieved by algorithm π interacting with the bandit instance θ_i. Based on the construction of bandit instances, we have

$$R_{0,T} \geqslant \frac{\Delta}{2} \sum_{i=1}^{K} \mathbb{E}_0\left[N_i(T)\right], \tag{7.6}$$

and for any $i \in [K]$

$$R_{i,T} \geqslant \frac{\Delta}{2}\left(T - \mathbb{E}_i[N_i(T)]\right) = \frac{T\Delta}{2}\left(1 - \frac{\mathbb{E}_i[N_i(T)]}{T}\right). \tag{7.7}$$

According to Lemma 7.12 and the calculation of KL-divergence between two Gaussian distributions, we further have

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}_{0,T}, \mathbb{P}_{i,T}) &= \mathbb{E}_0\left[\sum_{t=1}^{T} \mathrm{KL}\left(\mathcal{N}(\theta_0^\top A_t, 1/4), \mathcal{N}\left(\theta_i^\top A_t, 1/4\right)\right)\right] \\
&= \mathbb{E}_0\left[\sum_{t=1}^{T} 2\left\langle \theta_i - \theta_0, A_t\right\rangle^2\right] \\
&= 2\mathbb{E}_0\left[N_i(T)\right]\Delta^2, \qquad (7.8)
\end{aligned}$$

where Eq. (7.8) comes from the fact that $\theta_i = \theta_0 + \Delta \cdot e_{\rho(i)}$ and the only arm in $\mathcal{A}$ with non-zero value on the $\rho(i)$-th coordinate is $a_i = e_{\rho(i)}$, with $\langle \theta_i - \theta_0, a_i\rangle = \Delta$.

We now consider the average regret over $i \in [K]$:

$$\begin{aligned}
\frac{1}{K}\sum_{i=1}^{K} R_{i,T} &\geqslant \frac{T\Delta}{2}\left(1 - \frac{1}{K}\sum_{i=1}^{K}\frac{\mathbb{E}_i[N_i(T)]}{T}\right) \\
&\geqslant \frac{T\Delta}{2}\left(1 - \frac{1}{K}\sum_{i=1}^{K}\left(\frac{\mathbb{E}_0[N_i(T)]}{T} + \sqrt{\frac{\mathrm{KL}(\mathbb{P}_{i,T}, \mathbb{P}_{0,T})}{2}}\right)\right) \qquad (7.9) \\
&= \frac{T\Delta}{2}\left(1 - \frac{1}{K}\frac{\sum_{i=1}^{K}\mathbb{E}_0[N_i(T)]}{T} - \frac{1}{K}\sum_{i=1}^{K}\sqrt{\mathbb{E}_0\left[N_i(T)\right]\Delta^2}\right) \qquad (7.10) \\
&\geqslant \frac{T\Delta}{2}\left(1 - \frac{1}{K} - \sqrt{\frac{\sum_{i=1}^{K}\mathbb{E}_0\left[N_i(T)\right]\Delta^2}{K}}\right) \qquad (7.11) \\
&\geqslant \frac{T\Delta}{2}\left(1 - \frac{1}{K} - \sqrt{\frac{2\Delta R_{0,T}}{K}}\right) \qquad (7.12) \\
&\geqslant \frac{T\Delta}{2}\left(\frac{1}{2} - \sqrt{\frac{2\Delta B}{K}}\right), \qquad (7.13)
\end{aligned}$$

where Eq. (7.9) comes from applying Lemma 7.13 with $Z = N_i(T)/T$ and $\mathbb{P} = \mathbb{P}_{i,T}$ and $\mathbb{Q} = \mathbb{P}_{0,T}$; Eq. (7.10) comes from Lemma 7.12; Eq. (7.11) comes from concavity of $\sqrt{\cdot}$; Eq. (7.12) comes from Eq. (7.6); and finally Eq. (7.13) comes from the fact that $K \geqslant 2$ by construction and the assumption that $R_{0,T} \leqslant B$.

To obtain a large value for Eq. (7.13), one could maximize Δ while still make sure $\sqrt{2\Delta B/K} \leqslant 1/4$. Set $\Delta = 2^{-5}KB^{-1}$, following Eq. (7.13), we obtain

$$\begin{aligned}\frac{1}{K}\sum_{i=1}^{K} R_{i,T} &\geqslant 2^{-8}TKB^{-1} \\ &= 2^{-8}T\left\lfloor T^{\alpha}/2\right\rfloor B^{-1} \qquad (7.14)\\ &\geqslant 2^{-10}T^{1+\alpha}B^{-1}, \qquad (7.15)\end{aligned}$$

where Eq. (7.14) comes from the construction of K; and Eq. (7.15) comes from the assumption that $\lfloor T^{\alpha}/2\rfloor \geqslant T^{\alpha}/4$.

It is clear that any action $a \in \mathcal{A}$ satisfies $\|a\| \leqslant 1$ by construction, we now only need to make sure that $\|\theta_i\| \leqslant 1$ as well. Notice that $\|\theta_i\| \leqslant \sqrt{5}\Delta/2$ by construction, we only need to make sure $\Delta = 2^{-5}KB^{-1} \leqslant 2/\sqrt{5}$. Since on one hand $K = \lfloor T^{\alpha}/2\rfloor \leqslant T^{\alpha}$, and on the other hand $T^{\alpha} \leqslant B$ by assumption, we have $\Delta = 2^{-5}KB^{-1} \leqslant 2^{-5} < 2/\sqrt{5}$, as desired. □

7.7.1.2 Proof of Theorem 7.4

Lemma 7.14. *Suppose an algorithm achieves rate function $\theta(\alpha)$ on $\mathcal{H}_T(\alpha)$, then for any $0 < \alpha \leqslant 1$ such that $\alpha \leqslant \theta(0)$, we have*

$$\theta(\alpha) \geqslant 1 + \alpha - \theta(0). \qquad (7.16)$$

Proof. Fix $0 \leqslant \alpha \leqslant \theta(0)$. For any $\varepsilon > 0$, there exists constant c_1 and c_2 such that

$$\sup_{\omega \in \mathcal{H}_T(0)} R_T \leqslant c_1 T^{\theta(0)+\varepsilon} \quad \text{and} \quad \sup_{\omega \in \mathcal{H}_T(\alpha)} R_T \leqslant c_2 T^{\theta(\alpha)+\varepsilon},$$

for sufficiently large T. Let $B = \max\{c_1, 1\} \cdot T^{\theta(0)+\varepsilon}$, we could see that $T^{\alpha} \leqslant T^{\theta(0)} \leqslant B$

holds by assumption. For T large enough, the condition $\lfloor T^{\alpha}/2 \rfloor \geqslant \max\{T^{\alpha}/4, T^{0}, 2\}$ of Theorem 7.1 holds, and we then have

$$c_2 T^{\theta(\alpha)+\varepsilon} \geqslant 2^{-10} T^{1+\alpha} \left(\max\{c_1, 1\} \cdot T^{\theta(0)+\varepsilon}\right)^{-1} = 2^{-10} T^{1+\alpha-\theta(0)-\varepsilon} / \max\{c_1, 1\}.$$

For T sufficiently large, we then must have

$$\theta(\alpha) + \varepsilon \geqslant 1 + \alpha - \theta(0) - \varepsilon.$$

Let $\varepsilon \to 0$ leads to the desired result. □

Theorem 7.4. *Suppose a rate function θ is achieved by an algorithm, then we must have*

$$\theta(\alpha) \geqslant \min\{\max\{\theta(0), 1 + \alpha - \theta(0)\}, 1\}, \tag{7.2}$$

with $\theta(0) \in [1/2, 1]$.

Proof. For any adaptive rate function θ achieved by an algorithm, we first notice that $\theta(\alpha) \geqslant \theta(\alpha')$ for any $0 \leqslant \alpha' \leqslant \alpha \leqslant 1$ as $\mathcal{H}_T(\alpha') \subseteq \mathcal{H}_T(\alpha)$, which also implies $\theta(\alpha) \geqslant \theta(0)$. From Lemma 7.14, we further obtain $\theta(\alpha) \geqslant 1+\alpha-\theta(0)$ if $0 < \alpha \leqslant \theta(0)$. Thus, for any $\alpha \in (0, \theta(0)]$, we have

$$\theta(\alpha) \geqslant \max\{\theta(0), 1 + \alpha - \theta(0)\}. \tag{7.17}$$

Note that this indicates $\theta(\theta(0)) = 1$ since we trivially have $R_T \leqslant T$. For any $\alpha \in [\theta(0), 1]$, we have $\theta(\alpha) \geqslant \theta(\theta(0)) = 1$, which also leads to $\theta(\alpha) = 1$ for $\alpha \in [\theta(0), 1]$. To summarize, we obtain the desired result in Eq. (7.2). We have $\theta(0) \in [1/2, 1]$ as the minimax optimal rate among problems in $\mathcal{H}_T(0)$ is $1/2$ (Chu et al., 2011). □

7.7.2 Proofs and Supporting Results for Section 7.4

7.7.2.1 The virtual-mixture arm

The expected reward of virtual mixture-arm $\widetilde{\nu}_j$ can be expressed as the total expected reward obtained in iteration j divided by the corresponding time horizon ΔT_j:

$$\widetilde{\mu}_j = \mathbb{E}[\widetilde{\nu}_j] = \mathbb{E}\left[\sum_{t \text{ in iteration } j} X_t\right] / \Delta T_j = \langle \theta_\star, a_\star \rangle - R_{\Delta T_j} / \Delta T_j \in [-1, 1], \tag{7.18}$$

where we use $R_{\Delta T_j}$ to denote the expected regret suffered in iteration j. Let X_t be the reward obtained by pulling the virtual arm $\widetilde{\nu}_j$ (with A_t being the feature representation of the drawn real arm), we then know that $X_t - \widetilde{\mu}_j$ is $\sqrt{2}$-sub-Gaussian since $X_t - \widetilde{\mu}_j = (X_t - \langle \theta_\star, A_t \rangle) + (\langle \theta_\star, A_t \rangle - \widetilde{\mu}_j) = \eta_t + (\langle \theta_\star, A_t \rangle - \widetilde{\mu}_j)$: η_t is 1-sub-Gaussian by assumption and $(\langle \theta_\star, A_t \rangle - \widetilde{\mu}_j)$ is 1-sub-Gaussian due to boundedness $\langle \theta_\star, A_t \rangle \in [-1, 1]$ and $\mathbb{E}[\langle \theta_\star, A_t \rangle] = \widetilde{\mu}_j$.

7.7.2.2 Modifications of LinUCB

Recall that, under Assumption 7.6, the linear reward structure is preserved in the modified linear bandit problem that LinUCB will be working on in Algorithm 22. Two main differences in the modified linear bandit problem from the original setting considered in Chu et al. (2011) are: (1) we will be working with $\sqrt{2}$-sub-Gaussian noise while they deal with strictly bounded noise; and (2) the norm of our reward parameter, i.e., $\|\theta_\star^{\langle d_i \rangle}\|$, could be as large as $1 + (p-1) = p = \lceil \log_2(T^\beta) \rceil \leqslant \log_2(T) + 1 \leqslant 2 \log T$ when $T \geqslant 2$.

To reduce clutters, we consider a d dimensional linear bandit with time horizon T and K actions. We consider the reward structure $X_t = \langle \theta_\star, A_t \rangle + \eta_t$, where η_t is an independent $\sqrt{2}$-sub-Gaussian noise, $\|\theta_\star\| \leqslant 2 \log T$ and $\|A_t\| \leqslant 1$. The following Theorem 7.15 takes care of these changes.

Theorem 7.15. *For the modified setting introduced above, run LinUCB with*

$\alpha = 2\sqrt{\log(2TK/\delta)}$ *leads to an upper bound*

$$O\left(\log^2(KT\log(T)/\delta)\cdot\sqrt{dT}\right)$$

on the (pseudo) random regret with probability at least $1-\delta$.

Corollary 7.16. *For the modified setting introduced above, run LinUCB with* $\alpha = 2\sqrt{\log(2T^{3/2}K)}$ *leads to an upper bound*

$$O\left(\log^2(KT\log(T))\cdot\sqrt{dT}\right)$$

on the expected regret.

Proof. One can simply combine the result in Theorem 7.15 with $\delta = 1/\sqrt{T}$. □

It turns out that in order to prove Theorem 7.15, we mainly need to modify Lemma 1 in Chu et al. (2011), and the rest of the arguments go through smoothly. The changed exponent on the logarithmic term is due to $\|\theta_\star\| \leqslant 2\log T$. We introduce the following notations. Let

$$V_0 = I \quad \text{and} \quad V_t = V_{t-1} + A_t A_t^\top$$

denote the design matrix up to time t; and let

$$\widehat{\theta}_t = V_t^{-1}\sum_{i=1}^{t} A_i X_i$$

denote the estimate of $\theta_\star$ at time t.

Lemma 7.17. (*modification of Lemma 1 in Chu et al. (2011)*) *Suppose for any fixed sequence of selected actions* $\{A_i\}_{i\leqslant t}$ *the (random) rewards* $\{X_i\}_{i\leqslant t}$ *are independent. Then we have*

$$\mathbb{P}\left(\forall A_{t+1}\in\mathcal{A}_{t+1} : |\langle\widehat{\theta}_t - \theta_\star, A_{t+1}\rangle| \leqslant (\alpha + 2\log T)\sqrt{A_{t+1}^\top V_t^{-1} A_{t+1}}\right) \geqslant 1-\delta/T. \tag{7.19}$$

Remark 7.18. *The requirement of (conditional) independence is guaranted by the SupLinUCB algorithm introduced in Chu et al. (2011), and is not satisfied by the vanilla LinUCB: the reveal/selection of a future arm A_{t+1} makes previous rewards $\{X_i\}_{i\leqslant t}$ dependent. See Remark 4 in Han et al. (2020) for a detailed discussion.*

Proof. For any fixed A_t, we first notice that

$$\begin{aligned}
\left|\langle\widehat{\theta}_t - \theta_\star, A_{t+1}\rangle\right| &= |A_{t+1}^\top V_t^{-1} \sum_{i=1}^t A_i X_i - A_{t+1}^\top \theta_\star| \\
&= \left|A_{t+1}^\top V_t^{-1} \sum_{i=1}^t A_i X_i - A_{t+1}^\top V_t^{-1} \left(I + \sum_{i=1}^t A_i A_i^\top\right) \theta_\star\right| \\
&\leqslant \left|\sum_{i=1}^t A_{t+1}^\top V_t^{-1} A_i \left(X_i - A_i^\top \theta_\star\right)\right| + |A_{t+1}^\top V_t^{-1} \theta_\star| \\
&\leqslant \left|\sum_{i=1}^t A_{t+1}^\top V_t^{-1} A_i \left(X_i - A_i^\top \theta_\star\right)\right| + \|A_{t+1}^\top V_t^{-1}\| \cdot \|\theta_\star\|. \qquad (7.20)
\end{aligned}$$

We next bound the two terms in Eq. (7.20) seperately.

For the first term in Eq. (7.20), since $\left(X_i - A_i^\top \theta_\star\right)$ is $\sqrt{2}$-sub-Gaussian and $\{X_i\}_{i\leqslant t}$ are independent, we know that $\sum_{i=1}^t A_{t+1}^\top V_t^{-1} A_i \left(X_i - A_i^\top \theta_\star\right)$ is $\left(\sqrt{2\sum_{i=1}^t \left(A_{t+1}^\top V_t^{-1} A_i\right)^2}\right)$-sub-Gaussian. Since

$$\begin{aligned}
\sqrt{\sum_{i=1}^t \left(A_{t+1}^\top V_t^{-1} A_i\right)^2} &= \sqrt{\sum_{i=1}^t A_{t+1}^\top V_t^{-1} A_i A_i^\top V_t^{-1} A_{t+1}} \\
&\leqslant \sqrt{A_{t+1}^\top V_t^{-1} \left(I + \sum_{i=1}^t A_i A_i^\top\right) V_t^{-1} A_{t+1}} \\
&= \sqrt{A_{t+1}^\top V_t^{-1} A_{t+1}},
\end{aligned}$$

according to a standard Chernoff-Hoeffding bound, we have

$$\mathbb{P}\left(\left|\sum_{i=1}^{t} A_{t+1}^{\top} V_t^{-1} A_i \left(X_i - A_i^{\top}\theta_\star\right)\right| \geqslant \alpha\sqrt{A_{t+1}^{\top} V_t^{-1} A_{t+1}}\right) \leqslant 2\exp\left(-\frac{\alpha^2}{4}\right) = \frac{\delta}{TK}, \tag{7.21}$$

where Eq. (7.21) is due to $\alpha = 2\sqrt{\log(2TK/\delta)}$.

For the second term in Eq. (7.20), we have

$$\begin{aligned} \|A_{t+1}^{\top} V_t^{-1}\| \cdot \|\theta_\star\| &\leqslant 2\log T \sqrt{A_{t+1}^{\top} V_t^{-1} I V_t^{-1} A_{t+1}} \qquad (7.22)\\ &\leqslant 2\log T \sqrt{A_{t+1}^{\top} V_t^{-1}\left(I + \sum_{i=1}^{t} A_i A_i^{\top}\right) V_t^{-1} A_{t+1}} \\ &= 2\log T \sqrt{A_{t+1}^{\top} V_t^{-1} A_{t+1}}. \end{aligned}$$

where Eq. (7.22) comes from the fact that $\|\theta_\star\| \leqslant 2\log T$.

The desired result in Eq. (7.19) follows from a union bound argument together with the two upper bounds derived above. □

Remark 7.19. *Technically, regret guarantees are for a more complicated version of LinUCB that ensures statistical independence (Chu et al., 2011). However, as recommended by Chu et al. (2011), we will use the more practical LinUCB as our subroutine.*

7.7.2.3 Notations and Preliminaries for Analysis of LinUCB++

We provide some notations and preliminaries for analysis of LinUCB++ that will be used in the following two subsections, i.e., the proofs of Lemma 7.20 and Theorem 7.7.

We define $T_i = \sum_{j=1}^{i} \Delta T_j$ so that the i-th iteration of LinUCB++ goes from $T_{i-1}+1$ to T_i. We first notice that Algorithm 22 is a valid algorithm in the sense that it selects an arm A_t for any $t \in [T]$, i.e., it does not terminate before time T: the argument is

clearly true if there exists $i \in [p]$ such that $\Delta T_i = T$; otherwise, we can show that

$$T_p = \sum_{i=1}^{p} \Delta T_i = 2(2^{2p} - 1) \geqslant 2^{2p} \geqslant T,$$

for all $\beta \in [1/2, 1]$.

We use $R_{\Delta T_i} = \Delta T_i \cdot \mu_\star - \mathbb{E}[\sum_{t=T_{i-1}+1}^{T_i} X_t]$ to denote the expected cumulative regret at iteration i. Let $\mathcal{F}_i$ denote the information collected up to the end of iteration i, we further use $R_{\Delta T_i|\mathcal{F}_{i-1}}$ to represent the expected regret conditioned on $\mathcal{F}_{i-1}$ and have $\mathbb{E}[R_{\Delta T_i|\mathcal{F}_{i-1}}] = R_{\Delta T_i}$.

In the modified linear bandit problem at each iteration i, we will be applying LinUCB with respect to a $d_i + i - 1$ dimensional problem with an action set $\mathcal{A}^{\langle d_i \rangle}$ such that $\left|\mathcal{A}^{\langle d_i \rangle}\right| \leqslant K + i - 1$. Let $a_\star^{\langle d_i \rangle} = \arg\max_{a \in \mathcal{A}^{\langle d_i \rangle}} \{\langle \theta_\star^{\langle d_i \rangle}, a \rangle\}$ denote the best arm in the i-th iteration. Applying Eq. (7.3) on $R_{\Delta T_i|\mathcal{F}_{i-1}}$ leads to

$$R_{\Delta T_i|\mathcal{F}_{i-1}} = \Delta T_i \cdot \left(\langle \theta_\star, a_\star \rangle - \langle \theta_\star^{\langle d_i \rangle}, a_\star^{\langle d_i \rangle} \rangle\right) + \mathbb{E}\left[\sum_{t=T_{i-1}+1}^{T_i} \langle \theta_\star^{\langle d_i \rangle}, a_\star^{\langle d_i \rangle} - A_t \rangle \,\middle|\, \mathcal{F}_{i-1}\right], \tag{7.23}$$

where $A_t \in \mathcal{A}^{\langle d_i \rangle}$ and $\langle \theta_\star^{\langle d_i \rangle}, A_t \rangle$ represents the expected reward of pulling arm A_t.

7.7.2.4 Proof of Lemma 7.20

The proof of Lemma 7.20 follows the notations and preliminaries introduced in Section 7.7.2.3.

Lemma 7.20. *At each iteration $i \in [p]$, the learning error suffered from subroutine LinUCB is upper bounded by $O\left(\log^{5/2}(KT \log T) \cdot T^\beta\right)$.*

Proof. We focus on the second term in Eq. (7.23), i.e., the (conditional) learning error during iteration i. Conditioning on $\mathcal{F}_{i-1}$, both $\theta_\star^{\langle d_i \rangle}$ and $a_\star^{\langle d_i \rangle}$ can be treated

as fixed quantities. Applying the regret bound in Corollary 7.16, we have:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=T_{i-1}+1}^{T_i}\langle\theta_\star^{\langle d_i\rangle}, a_\star^{\langle d_i\rangle}-A_t\rangle\,\middle|\,\mathcal{F}_{i-1}\right]\\
&=O\left(\log^2\left((K+i-1)\Delta T_i\log(\Delta T_i)\right)\cdot\sqrt{(d_i+i-1)\Delta T_i}\right) &(7.24)\\
&=O\left(\log^2\left((K+p)\Delta T_i\log(\Delta T_i)\right)\cdot\sqrt{(d_i+p)\Delta T_i}\right) &(7.25)\\
&=O\left(\log^2\left((K+p)T\log T\right)\cdot\sqrt{2^{2p+2}+pT}\right) &(7.26)\\
&=O\left(\log^2\left(KT\log T\right)\cdot\sqrt{T^{2\beta}+\log T\cdot T}\right) &(7.27)\\
&=O\left(\log^{5/2}\left(KT\log T\right)\cdot T^{\beta}\right), &(7.28)
\end{aligned}$$

where Eq. (7.24) comes from the guarantee of LinUCB in Corollary 7.16; Eq. (7.25) uses the fact that $i\leqslant p$; Eq. (7.26) comes from the definition of d_i and ΔT_i; Eq. (7.27) comes from the fact that $p=\lceil\log_2 T^\beta\rceil$; Eq. (7.28) comes from trivially bounding $\sqrt{T^{2\beta}+\log T\cdot T}=O((\log T)^{1/2}\cdot T^\beta)$.[6] The desired result follows from taking another expectation over randomness in $\mathcal{F}_{i-1}$. □

7.7.2.5 Proof of Theorem 7.7

The proof of Theorem 7.7 follows the notations and preliminaries introduced in Section 7.7.2.3.

Theorem 7.7. *Run LinUCB++ with time horizon T and any user-specified parameter $\beta\in[1/2,1)$ leads to the following upper bound on the expected regret:*

$$\begin{aligned}
&\sup_{\omega\in\mathcal{H}_T(\alpha)} R_T\\
&=O\left(\log^{7/2}\left(KT\log T\right)\cdot T^{\min\{\max\{\beta,1+\alpha-\beta\},1\}}\right).
\end{aligned}$$

[6] One can improve the bound to $\sqrt{T^{2\beta}+\log T\cdot T}=O(T^\beta)$ in many cases, e.g., when $\beta>1/2$. We mainly focus on the polynomial terms here.

Proof. When $\alpha \geqslant \beta$, one could see that Theorem 7.7 trivially holds since $T^{1+\alpha-\beta} \geqslant T$. In the following, we only consider the case when $\alpha < \beta$.

Taking expectation on Eq. (7.23) and combining the result in Lemma 7.20, we obtain

$$R_{\Delta T_i} = \Delta T_i \cdot \mathbb{E}\left[\left(\langle \theta_\star, a_\star \rangle - \langle \theta_\star^{\langle d_i \rangle}, a_\star^{\langle d_i \rangle} \rangle\right)\right] + O\left(\log^{5/2}(KT \log T) \cdot T^\beta\right). \tag{7.29}$$

We now focus on the first term, i.e., the expected approximation error over the i-th iteration. Notice that, according to the definition of $a_\star^{\langle d_i \rangle}$ and $\theta_\star^{\langle d_i \rangle}$, we have $\langle \theta_\star^{\langle d_i \rangle}, a_\star^{\langle d_i \rangle} \rangle = \langle \theta_\star, a_\star \rangle$ if $d_i \geqslant d_\star$, i.e., the optimal arm is contained in the action set $\mathcal{A}^{\langle d_i \rangle}$. Let $i_\star \in [p]$ be the largest integer such that $d_{i_\star} \geqslant d_\star$, we then have that, for any $i \leqslant i_\star$ and in particular for $i = i_\star$,

$$R_{\Delta T_i} = O\left(T^\beta \log^{5/2}(KT \log T)\right). \tag{7.30}$$

In the case when $\Delta T_{i_\star} = \min\{2^{p+i_\star}, T\} = T$ or $i_\star = p$, we know that LinUCB++ will in fact stop at a time step no larger than $T_{i_\star}$ (since the allowed time horizon is T), and incur no regret in iterations $i > i_\star$. In the following, we only consider the case when $\Delta T_{i_\star} = 2^{p+i_\star}$ and $i_\star < p$. To incooperate another possible corner case when $d_{i_\star} = \min\{2^{p+2-i_\star}, d\} = d$, we consider $d_{i_\star+1} = 2^{p+1-i_\star} < d_{i_\star}$. As a result, we have $d_{i_\star} \Delta T_{i_\star} > d_{i_\star+1} \Delta T_{i_\star} = 2^{2p+1}$, which leads to

$$\Delta T_{i_\star} > \frac{2^{2p+1}}{d_{i_\star}} > \frac{2^{2p}}{d_\star} = \frac{2^{2p}}{T^\alpha}, \tag{7.31}$$

where Eq. (7.31) comes from the fact that $d_{i_\star} < 2d_\star$ according to the definition of $i_\star$.[7]

We now analysis the expected approximation error for iteration $i > i_\star$. Since the sampling information during $i_\star$-th iteration is summarized in the virtual mixture-arm $\widetilde{\nu}_{i_\star}$, and its representation $\widetilde{\nu}_{i_\star}^{\langle d_i \rangle}$ is added to $\mathcal{A}^{\langle d_i \rangle}$. For any $i > i_\star$, we then

[7] We will have $\Delta T_{i_\star} \geqslant 2^{2p+1}/T^\alpha > 2^{2p}/T^\alpha$ if $d_{i_\star} = \min\{2^{p+2-i_\star}, d\} = 2^{p+2-i_\star}$.

have

$$\begin{aligned}
\Delta T_i \cdot \mathbb{E}\left[\left(\langle \theta_\star, a_\star\rangle - \langle \theta_\star^{\langle d_i\rangle}, a_\star^{\langle d_i\rangle}\rangle\right)\right] &\leqslant \Delta T_i \cdot \mathbb{E}\left[\left(\langle \theta_\star, a_\star\rangle - \langle \theta_\star^{\langle d_i\rangle}, \widetilde{\nu}_{i_\star}^{\langle d_i\rangle}\rangle\right)\right] \\
&= \Delta T_i \cdot (\langle \theta_\star, a_\star\rangle - \widetilde{\mu}_{i_\star}) && (7.32)\\
&= \frac{\Delta T_i}{\Delta T_{i_\star}} \cdot R_{\Delta T_{i_\star}} && (7.33)\\
&= \frac{\Delta T_i}{\frac{2^{2p}}{T^\alpha}} \cdot O\left(\log^{5/2}(KT\log T)\cdot T^\beta\right) && (7.34)\\
&= \frac{O\left(\log^{5/2}(KT\log T)\cdot T^{1+\alpha+\beta}\right)}{2^{2p}} && (7.35)\\
&= O\left(\log^{5/2}(KT\log T)\cdot T^{1+\alpha-\beta}\right), && (7.36)
\end{aligned}$$

where Eq. (7.32) comes from the formulation of the modified linear bandit problem; Eq. (7.33) comes from that fact that $\widetilde{\mu}_j = \mathbb{E}[\widetilde{\mu}_{j|\mathcal{F}_j}] = \langle \theta_\star, a_\star\rangle - R_{\Delta T_j}/\Delta T_j$ derived from Eq. (7.18); Eq. (7.34) comes from the bound in Eq. (7.30) with $i = i_\star$; Eq. (7.35) comes from the fact that $\Delta T_i \leqslant T$ and some rewriting; Eq. (7.36) comes from the fact that $p = \lceil \log_2 T^\beta \rceil \geqslant \log_2 T^\beta$.

Combining Eq. (7.36) and Eq. (7.29) for cases when $i > i_\star$ (or the corner case algorithm stops before $T_{i_\star}$ and incurs no regret in iterations $i \geqslant i_\star$), and together with Eq. (7.30) for cases when $i \leqslant i_\star$, we have that $\forall i \in [p]$,

$$R_{\Delta T_i} = O\left(\log^{5/2}(KT\log T)\cdot T^{\max\{\beta,1+\alpha-\beta\}}\right).$$

Since the cumulative regret is non-decreasing in t, we have

$$\begin{aligned}
R_T &\leqslant \sum_{i=1}^{p} R_{\Delta T_i} \\
&= \sum_{i=1}^{p} O\left(\log^{5/2}(KT\log T)\cdot T^{\max\{\beta,1+\alpha-\beta\}}\right) \\
&= O\left(\log^{7/2}(KT\log T)\cdot T^{\max\{\beta,1+\alpha-\beta\}}\right),
\end{aligned}$$

where we use the fact that $p = \lceil \log_2(T^\beta) \rceil = O(\log T)$. Our results follows after noticing $R_T \leqslant T$ is a trivial upper bound. □

7.7.2.6 Proof of Theorem 7.8

Theorem 7.8. *The rate function achieved by LinUCB++ with any input $\beta \in [1/2, 1)$, i.e.,*

$$\theta_\beta : \alpha \mapsto \min\{\max\{\beta, 1 + \alpha - \beta\}, 1\}, \tag{7.4}$$

is Pareto optimal.

Proof. From Theorem 7.7, we know that the rate in Eq. (7.4) is achieved by Algorithm 22 with input β. We only need to prove that no other algorithms achieve strictly smaller rates in pointwise order.

Suppose, by contradiction, we have θ' achieved by an algorithm such that $\theta'(\alpha) \leqslant \theta_\beta(\alpha)$ for all $\alpha \in [0, 1]$ and $\theta'(\alpha_0) < \theta(\alpha_0)$ for at least one $\alpha_0 \in [0, 1]$. We then must have $\theta'(0) \leqslant \theta_\beta(0) = \beta$. We consider the following two exclusive cases.

Case 1: $\theta'(0) = \beta$**.** According to Theorem 7.4, we must have $\theta' \geqslant \theta_\beta$, which leads to a contradiction.

Case 2: $\theta'(0) = \beta' < \beta$**.** According Theorem 7.4, we must have $\theta' \geqslant \theta_{\beta'}$. However, $\theta_{\beta'}$ is not strictly better than θ_β, e.g., $\theta_{\beta'}(2\beta - 1) = 2\beta - \beta' > \beta = \theta_\beta(2\beta - 1)$, which also leads to a contradiction. □

7.7.3 Proofs and Supporting Results for Section 7.4.2

7.7.3.1 Discussion on Algorithm 23

We construct the following two (smoothed) base algorithms (Pacchiano et al., 2020b) at each iteration of LinUCB++: (1) a LinUCB algorithm that works with truncated feature representations in $\mathbb{R}^{d_i}$, with possible mis-specifications; and (2) a UCB algorithm that works only with virtual mixture-arms, if there exists any. We use Smooth Corral from Pacchiano et al. (2020b) as the master algorithm and

always optimally tune it with respect to the LinUCB base, i.e., set the learning rate as $\eta = 1/\sqrt{d_i \Delta T_i}$. For iterations such that $d_i \geqslant d_\star$, the LinUCB is the optimal base and we incur $\widetilde{O}(\sqrt{d_i \Delta T_i}) = \widetilde{O}(T^\beta)$ regret; a good enough virtual mixture-arm $\widetilde{\nu}_{i_\star}$ is then constructed as before. For later iterations such that $d_i < d_\star$, Smooth Corral incurs regret $\widetilde{O}(\max\{T^{1+\alpha-\beta}, T^\beta\})$ thanks to guarantees of the UCB base: the $\widetilde{O}(T^{1+\alpha-\beta})$ term is due to the approximation error and the $\widetilde{O}(T^\beta)$ term is due to the learning error. Although the learning error of UCB is enlarged from $\widetilde{O}(T^{1/2})$ to $\widetilde{O}(T^\beta)$, as Smooth Corral is always tuned with respect to the LinUCB base, this won't affect the resulted Pareto optimality.

7.7.3.2 Proof of Theorem 7.9

Theorem 7.9. *With any input $\beta \in [1/2, 1)$, the rate function achieved by Algorithm 23 (without Assumption 7.6) is Pareto optimal.*

Proof. At each iteration $i \in [p]$ of LinUCB++, we applying Smooth Corral as the master algorithm with two smoothed base algorithms: (1) a LinUCB algorithm that works with truncated feature representations in $\mathbb{R}^{d_i}$, with possible mis-specifications; and (2) a UCB algorithm that works only with virtual mixture-arms, if there exists any. The learning rate of Smooth Corral is always optimally tuned with respect to the LinUCB base, i.e., $\eta = 1/\sqrt{d_i \Delta T_i}$. Since there are at most $p = O(\log T)$ iterations, we only need to bound the expected regret at each iteration $R_{\Delta T_i}$. As before, we use $i_\star \in [p]$ to denote the largest integer such that $d_{i_\star} \geqslant d_\star$.

For $i \leqslant i_\star$, the LinUCB base works on a well-specified linear bandit problem. Theorem 5.3 in Pacchiano et al. (2020b) gives the following guarantees:

$$R_{\Delta T_i} = \widetilde{O}\left(\sqrt{\Delta T_i} + \eta^{-1} + \Delta T_i \eta + \Delta T_i d_i \eta\right) = \widetilde{O}\left(\sqrt{d_i \Delta T_i}\right) = \widetilde{O}\left(T^\beta\right).$$

Good enough virtual mixture-arm $\widetilde{\nu}_{i_\star}$ is then constructed with conditional expectation $\widetilde{\mu}_{i_\star|\mathcal{F}_{i_\star}} = \mathbb{E}[\widetilde{\nu}_{i_\star}|\mathcal{F}_{i_\star}] = \langle \theta_\star, a_\star \rangle - \widehat{R}_{\Delta T_{i_\star}}/\Delta T_{i_\star}$.

We now analyze the regret incurred for iteration $i > i_\star$. Conditioning on past information $\mathcal{F}_{i-1}$ and let $r(\pi_t)$ denote the (conditional) expected reward of applying

policy π_t, we have

$$\begin{aligned} R_{\Delta T_i|\mathcal{F}_{i-1}} &= \Delta T_i \cdot \left(\langle \theta_\star, a_\star \rangle - \widetilde{\mu}_{i_\star|\mathcal{F}_{i_\star}}\right) + \mathbb{E}\left[\sum_{t \text{ in iteration } i} \widetilde{\mu}_{i_\star|\mathcal{F}_{i_\star}} - r(\pi_t) \,\middle|\, \mathcal{F}_{i-1}\right] \\ &= \Delta T_i \cdot \left(\langle \theta_\star, a_\star \rangle - \widetilde{\mu}_{i_\star|\mathcal{F}_{i_\star}}\right) + \widetilde{O}\left(\sqrt{\Delta T_i} + \eta^{-1} + \Delta T_i \eta + \Delta T_i \eta\right), \end{aligned}$$

where the second term comes from the guarantee of Smooth Corral with respect to the UCB base. Taking expectation over randomness in $\mathcal{F}_{i-1}$ leads to

$$R_{\Delta T_i} = \widetilde{O}\left(T^{1+\alpha-\beta}\right) + \widetilde{O}\left(T^{\beta}\right),$$

where the first term follows from a similar analysis as in Eq. (7.36), and the second term follows by setting $\eta = 1/\sqrt{d_i \Delta T_i}$. A similar analysis as in Theorem 7.8 thus show Algorithm 23 is Pareto optimal, even without Assumption 7.6. □

7.7.3.3 Discussion on Smooth Corral

Pacchiano et al. (2020b) tackles the model selection problem in linear bandit by applying Smooth Corral with $O(\log d)$ base LinUCB learners working with different dimensions $d_i \in \{2^0, 2^1, \ldots, 2^{\lfloor \log d \rfloor}\}$. Let $d_{i_\star}$ denote the smallest dimension that satisfies $d_{i_\star} \geqslant d_\star$. With respect to the base LinUCB working on the first $d_{i_\star}$ dimensions, Smooth Corral enjoys regret guarantee

$$R_T = \widetilde{O}\left(\sqrt{T} + \eta^{-1} + T\eta + T d_\star \eta\right).$$

Smooth Corral then achieves the rate function in Eq. (7.4) by setting the learning rate $\eta = T^{-\beta}$ (and also noticing that $d_\star \leqslant T^\alpha$).

7.7.4 Other Details for Experiments

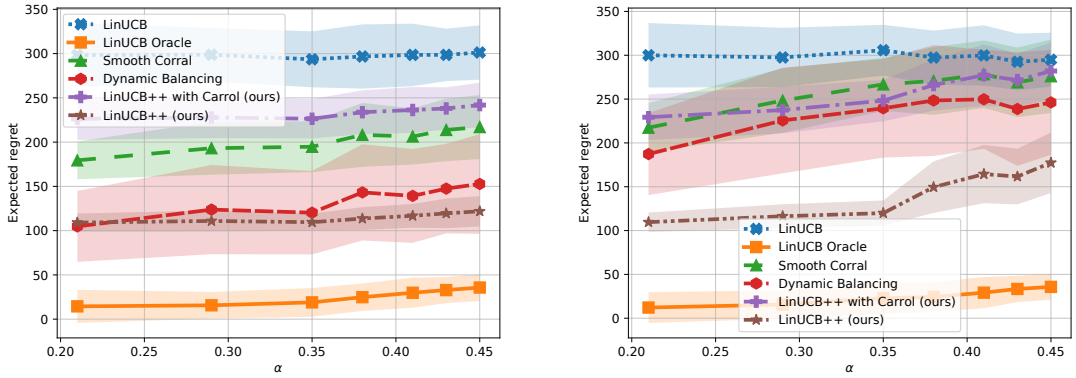


Figure 7.4: Similar experiment setups to those shown in Fig. 7.2b, but with different reward parameters $\theta_\star$.

We conduct additional experiments with setups similar to the ones shown in Fig. 7.2b, but with different reward parameters $\theta_\star$. We set $\theta_\star$ as (the normalized version of) $[\frac{1}{\sqrt{1}}, \frac{1}{\sqrt{2}}, \ldots, \frac{1}{\sqrt{d_\star}}, 0, \ldots, 0]^\top \in \mathbb{R}^d$ in Fig. 7.4a; and $\theta_\star$ as (the normalized version of) $[\frac{1}{\sqrt{d_\star}}, \frac{1}{\sqrt{d_\star - 1}}, \ldots, \frac{1}{\sqrt{1}}, 0, \ldots, 0]^\top \in \mathbb{R}^d$ in Fig. 7.4b. With $\theta_\star$ selected in Fig. 7.4a, Dynamic Balancing shows comparable performance to LinUCB++ in terms of averaged regret (but with larger variance). LinUCB++ outperforms Dynamic Balancing when $\theta_\star$ is "flipped" (i.e., the one used in Fig. 7.4b) but with the same intrinsic dimension $d_\star$.

8 MODEL SELECTION FOR BEST ACTION IDENTIFICATION

We introduce the model selection problem in pure exploration linear bandits, where the learner needs to adapt to the instance-dependent complexity measure of the smallest hypothesis class containing the true model. We design algorithms in both fixed confidence and fixed budget settings with near instance optimal guarantees. The core of our algorithms is a new optimization problem based on experimental design that leverages the geometry of the action set to identify a near-optimal hypothesis class. Our fixed budget algorithm is developed based on a novel selection-validation procedure, which provides a new way to study the understudied fixed budget setting (even without the added challenge of model selection). We adapt our algorithms, in both fixed confidence and fixed budget settings, to problems with model misspecification.

8.1 Introduction

The pure exploration linear bandit problem considers a set of arms whose expected rewards are linear in their *given* feature representation, and aims to identify the optimal arm through adaptive sampling. Two settings, i.e., fixed confidence and fixed budget settings, are studied in the literature. In the fixed confidence setting, the learner continues sampling arms until a desired confidence level is reached, and the goal is to minimize the total number of samples (Soare et al., 2014; Xu et al., 2018; Tao et al., 2018; Fiez et al., 2019; Degenne et al., 2020; Katz-Samuels et al., 2020). In the fixed budget setting, the learner is forced to output a recommendation within a prefixed sampling budget, and the goal is to minimize the error probability (Hoffman et al., 2014; Katz-Samuels et al., 2020; Alieva et al., 2021; Yang and Tan, 2021). Applications of pure exploration linear bandits include content recommendation, digital advertisement and A/B/n testing (see aforementioned papers for more discussions on applications).

All existing works, however, focus on linear models with the *given* feature

representations and fail to adapt to cases when the problem can be explained with a much simpler model, i.e., a linear model based on a subset of the features. In this chapter, we introduce the model selection problem in pure exploration linear bandits. We consider a sequence of nested linear hypothesis classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \subseteq \mathcal{H}_D$ and assume that $\mathcal{H}_{d_\star}$ is the smallest hypothesis class that contains the true model. Our goal is to automatically adapt to the complexity measure related to $\mathcal{H}_{d_\star}$, for an unknown $d_\star$, rather than suffering a complexity measure related to the largest hypothesis class $\mathcal{H}_D$.

The model selection problem appears ubiquitously in real-world applications. In fact, cross-validation (Stone, 1974, 1978), a practical method for model selection, appears in almost all successful deployments of machine learning models. The model selection problem was recently introduced to the bandit regret minimization setting by Foster et al. (2019), and further analyzed by Pacchiano et al. (2020b); Zhu and Nowak (2022c). Zhu and Nowak (2022c) prove that only Pareto optimality can be achieved for regret minimization, which is even weaker than minimax optimality. We introduce the model selection problem in the pure exploration setting and, surprisingly, show that it is possible to design algorithms with *near optimal instance-dependent complexity* for both fixed confidence and fixed budget settings. We further generalize the model selection problem to the regime with misspecified linear models, and show our algorithms are robust to model misspecification.

8.1.1 Contribution and Organization

We briefly summarize our contributions as follows:

- We introduce the model selection problem for pure exploration in linear bandits in Section 8.2, and analyze its instance-dependent complexity measure. We provide a general framework to solve the model selection problem for pure exploration linear bandits. Our framework is based on a carefully-designed two-dimensional doubling trick and a new optimization problem that leverages the geometry of the action set to efficiently identify a near-optimal hypothesis class.

- In Section 8.4, we provide an algorithm for the fixed confidence setting with near optimal instance-dependent unverifiable sample complexity. We additionally provide evidence on why one cannot verifiably output recommendations.
- In Section 8.5, we provide an algorithm for the fixed budget setting, which applies a novel selection-validation trick to bandits. Its probability of error matches (up to logarithmic factors) the probability error of an algorithm that chooses its sampling allocation based on knowledge of the true model parameter. In addition, the guarantee of our algorithm is nearly optimal even in the non-model-selection case, and our algorithm also provides a new way to analyze the *understudied* fixed budget setting.
- We further generalize the model selection problem into the misspecified regime in Section 8.6, and adapt our algorithms to both the fixed confidence and fixed budget settings. Our algorithms reach an instance-dependent sample complexity measure that is relevant to the complexity measure of a closely related perfect linear bandit problem.

8.2 Problem Setting

In the transductive linear bandit pure exploration problem, the learner is given an action set $\mathcal{X} \subset \mathbb{R}^D$ and a target set $\mathcal{Z} \subset \mathbb{R}^D$. The expected reward of any arm $x \in \mathcal{X} \cup \mathcal{Z}$ is linearly parameterized by an unknown reward vector $\theta_\star \in \Theta \subseteq \mathbb{R}^D$, i.e., $h(x) = \langle \theta_\star, x \rangle$. The parameter space Θ is known to the learner. At each round t, the learner/algorithm $\mathscr{A}$ selects an action $X_t \in \mathcal{X}$, and observes a noisy reward $R_t = h(X_t) + \xi_t$, where ξ_t represents an additive 1-sub-Gaussian noise. The action $X_t \in \mathcal{X}$ can be selected with respect to the history $\mathcal{F}_{t-1} = \sigma((X_i, R_i)_{i<t})$ up to time t. The goal is to identify the unique optimal arm $z_\star = \arg\max_{z \in \mathcal{Z}} h(z)$ from the target set $\mathcal{Z}$. We assume $\Theta \subseteq \operatorname{span}(\mathcal{X})$ to obtain unbiased estimators for arms in $\mathcal{Z}$. Without loss of generality, we assume that $\operatorname{span}(\mathcal{X}) = \mathbb{R}^D$ (otherwise one can project actions into a lower dimensional space). We further assume that $\operatorname{span}(\{z_\star - z\}_{z \in \mathcal{Z}}) = \mathbb{R}^D$

for technical reasons. We consider both fixed confidence and fixed budget settings in this chapter.

Definition 8.1 (Fixed confidence)**.** *Fix $\mathcal{X}, \mathcal{Z}, \Theta \subseteq \mathbb{R}^D$. An algorithm $\mathscr{A}$ is called δ-PAC for $(\mathcal{X}, \mathcal{Z}, \Theta)$ if (1) the algorithm has a stopping time τ with respect to $\{\mathcal{F}_t\}_{t\in\mathbb{N}}$ and (2) at time τ it makes a recommendation $\widehat{z} \in \mathcal{Z}$ such that $\mathbb{P}_{\theta_\star}(\widehat{z} = z_\star) \geqslant 1 - \delta$ for all $\theta_\star \in \Theta$.*

Definition 8.2 (Fixed budget)**.** *Fix $\mathcal{X}, \mathcal{Z}, \Theta \subseteq \mathbb{R}^D$ and a budget T. A fixed budget algorithm $\mathscr{A}$ returns a recommendation $\widehat{z} \in \mathcal{Z}$ after T rounds.*

The model selection problem. The learner is given a nested sequence of parameter classes $\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_D$, where $\Theta_d := \left\{\theta \in \mathbb{R}^D : \theta_i = 0, \forall i > d\right\}$ is the set of parameters such that for any $\theta \in \Theta_d$, it only has non-zero entries on its first d coordinates.[1] We assume that $\theta_\star \in \Theta_{d_\star}$ for an *unknown* $d_\star$. We call $d_\star$ the intrinsic dimension of the problem and it is set as the index of the smallest parameter space containing the true reward vector. One interpretation of the intrinsic dimension is that only the first $d_\star$ features (of each arm) play a role in predicting the expected reward. Our goal is to automatically adapt to the sample complexity with respect to the intrinsic dimension $d_\star$, rather than suffering from the sample complexity related to the ambient dimension D. In the following, we write $\theta_\star \in \Theta_{d_\star}$ to indicate that the problem instance has intrinsic dimension $d_\star$. Besides dealing with the *well-specified* linear bandit problem as defined in this section, we also extend our framework into the *misspecified* setting in Section 8.6, with additional setups introduced therein.

additional notations. For any $x = [x_1, x_2, \ldots, x_D]^\top \in \mathbb{R}^D$ and $d \leqslant D$, we use $\psi_d(x) := [x_1, x_2, \ldots, x_d]^\top \in \mathbb{R}^d$ to denote the truncated feature representation that only keeps its first d coordinates. We also write $\psi_d(\mathcal{X}) := \{\psi_d(x) : x \in \mathcal{X}\}$ and $\psi_d(\mathcal{Z}) := \{\psi_d(z) : z \in \mathcal{Z}\}$ to represent the truncated action set and target set, respectively. Note that we necessarily have $\psi_d(\mathcal{Z}) \subseteq \operatorname{span}(\psi_d(\mathcal{X})) = \mathbb{R}^d$ as long as $\mathcal{Z} \subseteq \operatorname{span}(\mathcal{X}) = \mathbb{R}^D$. We use $\mathcal{Y}(\psi_d(\mathcal{Z})) := \{\psi_d(z) - \psi_d(z') : z, z' \in \mathcal{Z}\}$

[1] A nested sequence of linear hypothesis classes $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \cdots \subseteq \mathcal{H}_D$ can be constructed based on the nested sequence of parameter classes $\Theta_1 \subseteq \Theta_2 \subseteq \cdots \subseteq \Theta_D$, i.e., $\mathcal{H}_d := \{h(\cdot) = \langle \theta, \cdot \rangle : \theta \in \Theta_d\}$.

to denote all possible directions formed by subtracted one item from another in $\psi_d(\mathcal{Z})$; and use $\mathcal{Y}^\star(\psi_d(\mathcal{Z})) := \{\psi_d(z_\star) - \psi_d(z) : z \in \mathcal{Z}\}$ to denote all possible directions with respect to the optimal arm $z_\star$. For any $z \in \mathcal{Z}$, we use $\Delta_z := h(z_\star) - h(z)$ to denote its sub-optimality gap; we set $\Delta_{\min} := \min_{z\in\mathcal{Z}\setminus\{z_\star\}} \Delta_z$. As in Fiez et al. (2019), we assume $\max_{z\in\mathcal{Z}} \Delta_z \leqslant 2$ when analyzing upper bounds. We denote $\mathcal{S}_k := \left\{z \in \mathcal{Z} : \Delta_z < 4 \cdot 2^{-k}\right\}$ (with $\mathcal{S}_1 := \mathcal{Z}$). We use $\Delta_\mathcal{X} = \Delta(\mathcal{X}) := \left\{\lambda \in \mathbb{R}^{|\mathcal{X}|} : \sum_{x\in\mathcal{X}} \lambda_x = 1, \lambda_x \geqslant 0\right\}$ to denote the $(|\mathcal{X}| - 1)$-dimensional simplex over actions. For any (continuous) design $\lambda \in \Delta_\mathcal{X}$, we use $A_d(\lambda) := \sum_{x\in\mathcal{X}} \lambda_x \psi_d(x)(\psi_d(x))^\top \in \mathbb{R}^{d\times d}$ to denote the design matrix with respect to λ. For any set $\mathcal{W} \subseteq \mathbb{R}^D$, we denote $\iota(\mathcal{W}) := \inf_{\lambda\in\Delta_\mathcal{X}} \sup_{w\in\mathcal{W}} \|w\|^2_{A_d(\lambda)^{-1}}$.[2]

8.3 Towards the True Sample Complexity

The instance-dependent sample complexity lower bound for linear bandit is discovered/analyzed in previous papers (Soare et al., 2014; Fiez et al., 2019; Degenne and Koolen, 2019). We here consider related quantities that take our model selection setting into consideration. For any $d \in [D]$, we define

$$\rho^\star_d := \inf_{\lambda\in\Delta_\mathcal{X}} \sup_{z\in\mathcal{Z}\setminus\{z_\star\}} \frac{\|\psi_d(z_\star) - \psi_d(z)\|^2_{A_d(\lambda)^{-1}}}{(h(z_\star) - h(z))^2}, \tag{8.1}$$

and

$$\iota^\star_d := \inf_{\lambda\in\Delta_\mathcal{X}} \sup_{z\in\mathcal{Z}\setminus\{z_\star\}} \|\psi_d(z_\star) - \psi_d(z)\|^2_{A_d(\lambda)^{-1}}. \tag{8.2}$$

Following analysis in Fiez et al. (2019), we provide a lower bound for the model selection problem $(\mathcal{X}, \mathcal{Z}, \theta_\star \in \Theta_{d_\star})$ in the fixed confidence setting as follows.

Theorem 8.3. *Suppose* $\xi_t \sim \mathcal{N}(0,1)$ *for all* $t \in \mathbb{N}_+$ *and* $\delta \in (0, 0.15]$. *Any* δ*-PAC algorithm with respect to* $(\mathcal{X}, \mathcal{Z}, \theta_\star \in \Theta_{d_\star})$ *with stopping time* τ *satisfies* $\mathbb{E}_{\theta_\star}[\tau] \geqslant \rho^\star_{d_\star} \log(1/2.4\delta)$.

[2] A generalized inversion is used for singular matrices. See Section 8.9.1.1 for detailed discussion.

The above lower bound only works for δ-PAC algorithms, but not for algorithms in the fixed budget setting or with unverifiable sample complexity (see Section 8.4). We now introduce another lower bound for the best possible *non-interactive* algorithm $\mathscr{A}$. Following the discussion in Katz-Samuels et al. (2020), we consider any non-interactive algorithm as follows: The algorithm $\mathscr{A}$ chooses an allocation $\{x_1, x_2, \ldots, x_N\} \subseteq \mathcal{X}$ and receive rewards $\{r_1, r_2, \ldots, r_N\} \subseteq \mathbb{R}$ where r_i is sampled from $\mathcal{N}(h(x_i), 1)$. The algorithm then recommends $\widehat{z} = \arg\max_{z \in \mathcal{Z}} \left\langle \widehat{\theta}_d, z \right\rangle$ where $\widehat{\theta}_d = \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^N (r_i - \theta^\top \psi_d(x_i))^2$ is the least squares estimator in $\mathbb{R}^d$. The learner is allowed to choose any allocations, *even with the knowledge of* $\theta_\star$, and use any feature mapping such that linearity is preserved, i.e., $d_\star \leqslant d \leqslant D$.

Theorem 8.4. *Fix* $(\mathcal{X}, \mathcal{Z}, \theta_\star \in \Theta_{d_\star})$ *and* $\delta \in (0, 0.015]$. *Any non-interactive algorithm* $\mathscr{A}$ *using a feature mappings of dimension* $d \geqslant d_\star$ *makes a mistake with probability at least* δ *as long as it uses no more than* $\frac{1}{2}\rho^\star_{d_\star} \log(1/\delta)$ *samples.*

The above lower bound serves as a fairly strong baseline due to the power provided to the non-interactive learner, i.e., the knowledge of $\theta_\star$. Theorem 8.4 indicates (for any non-interactive learner) (1) sample complexity lower bound $\widetilde{\Omega}(\rho^\star_{d_\star})$ in fixed confidence setting; and (2) error probability lower bound $\Omega(\exp(-T/\rho^\star_{d_\star}))$ in fixed budget setting: Suppose the budget is T, one would expect an error probability at least $\Omega(\exp(-T/\rho^\star_{d_\star}))$ by relating $\frac{1}{2}\rho^\star_{d_\star} \log(1/\delta)$ to T.

Note that all lower bounds are with respect to $\rho^\star_{d_\star}$ rather than $\rho^\star_d$ for $d > d_\star$ due to the assumption $\theta_\star \in \Theta_{d_\star}$ for the model selection problem. Our goal is to automatically adapt to the complexity $\rho^\star_{d_\star}$ without knowledge of $d_\star$. The following proposition shows the monotonic relation among $\{\rho^\star_d\}_{d=d_\star}^D$.

Proposition 8.5. *The monotonic relation* $\rho^\star_{d_1} \leqslant \rho^\star_{d_2}$ *holds true for any* $d_\star \leqslant d_1 \leqslant d_2 \leqslant D$.

The intuition behind Proposition 8.5 is that the model class Θ_{d_2} is a superset of Θ_{d_1} and therefore identifying $z_\star$ in Θ_{d_2} requires ruling out a larger set of statistical alternatives than in Θ_{d_1}. While Proposition 8.5 is intuitive, its proof is surprisingly technical and involves showing the equivalence of a series of optimization problems.

8.3.1 Failure of Standard Approaches

Proposition 8.6. *For any $\gamma > 0$, there exists an instance $(\mathcal{X}, \mathcal{Z}, \theta_\star \in \Theta_{d_\star})$ such that $\rho^\star_{d_\star+1} > \rho^\star_{d_\star} + \gamma$ yet $\iota^\star_{d_\star+1} \leqslant 2\iota^\star_{d_\star}$.*

One may attempt to solve the model selection problem with a standard doubling trick over dimension, i.e., truncating the feature representations at dimension $d_i = 2^i$ for $i \leqslant \lceil \log_2 D \rceil$ and gradually exploring models with increasing dimension. This approach, however, is directly ruled out by Proposition 8.6 since such doubling trick could end up with solving a problem with a dimension $d' \leqslant 2d_\star$ yet $\rho^\star_{d'} \gg \rho^\star_{d_\star}$. Although doubling trick over dimensions is commonly used to provide *worst-case* guarantees in regret minimization settings (Pacchiano et al., 2020b; Zhu and Nowak, 2022c), we emphasize here that matching *instance-dependent* complexities is important in pure exploration setting (Soare et al., 2014; Fiez et al., 2019; Katz-Samuels et al., 2020). Thus, new techniques need to be developed. Proposition 8.6 also implies that trying to infer the value of $\rho^\star_d$ from $\iota^\star_d$ can be quite misleading. And thus conducting a doubling trick over $\iota^\star_d$ (or an upper bound of it) is likely to fail as well.

Importance of model selection. Proposition 8.6 also illustrates the importance and necessity of conducting model selection in pure exploration linear bandits. Consider the hard instance used in constructed in Proposition 8.6 and set $D = d_\star + 1$. All existing algorithms (Soare et al., 2014; Fiez et al., 2019; Degenne and Koolen, 2019; Katz-Samuels et al., 2020) that directly work with the *given* feature representation in $\mathbb{R}^D$ end up with a complexity measure scales with $\rho^\star_D$, which could be arbitrarily large than the true complexity measure $\rho^\star_{d_\star}$ and even become vacuous (by sending $\gamma \to \infty$).

Our approaches. In this chapter, we design a more sophisticated doubling scheme over a two-dimensional grid corresponding to the number of elimination steps and the richest hypothesis class considered at each step. We design subroutines for both fixed confidence and fixed budget settings. Our algorithms define a new

optimization problem based on experimental design that leverages the geometry of the action set to efficiently identify a near-optimal hypothesis class. Our fixed budget algorithm additionally uses a novel application of a selection-validation trick in bandits. Our guarantees are with respect to the true instance-dependent complexity measure $\rho^{\star}_{d_\star}$.

8.4 Fixed Confidence Setting

We present our main algorithm (Algorithm 25) for the fixed confidence setting in this section. Algorithm 25 invokes GEMS-c (Algorithm 24) as subroutines and starts to output the optimal arm after $\widetilde{O}(\rho^{\star}_{d_\star} + d_\star)$ samples. Our sample complexity matches, up to an additive $d_\star$ term and logarithmic factors, the strong baseline developed in Theorem 8.4.

We first introduce the subroutine GEMS-c, which runs for n rounds and takes (roughly) B samples per-round. GEMS-c is built on RAGE (Fiez et al., 2019), a standard linear bandit pure exploration algorithm works in the ambient space $\mathbb{R}^D$. The key innovation of GEMS-c lies in *adaptive* hypothesis class selection at each round (i.e., selecting d_k), which allows us to adapt to the instrinsic dimension $d_\star$. After selecting the working dimension d_k at round k, GEMS-c allocates samples based on optimal design (in $\mathbb{R}^{d_k}$); it then eliminate sub-optimal arms based on the estimated rewards constructed using least squares. Following Fiez et al. (2019), we use a rounding procedure $\mathsf{ROUND}(\lambda, N, d, \zeta)$ to round a continuous experimental design $\lambda \in \Delta_{\mathcal{X}}$ into integer allocations over actions. We use $r_d(\zeta)$ to denote the number of samples needed for such rounding in $\mathbb{R}^d$ with approximation factor ζ. One can choose $r_d(\zeta) = (d^2 + d + 2)/\zeta$ (Pukelsheim, 2006; Fiez et al., 2019) or $r_d(\zeta) = 180d/\zeta^2$ (Allen-Zhu et al., 2020). We choose ζ as a constant throughout this chapter, e.g., $\zeta = 1$. When $N \geqslant r_d(\zeta)$, there exist computationally efficient rounding procedures that output an allocation $\{x_1, x_2, \ldots, x_N\}$ satisfying

$$\max_{y \in \mathcal{Y}(\psi_d(\mathcal{Z}))} \|y\|^2_{(\sum_{i=1}^N \psi_d(x_i)\psi_d(x_i)^\top)^{-1}} \leqslant$$

$$(1+\zeta)\max_{y\in\mathcal{Y}(\psi_d(\mathcal{Z}))}\|y\|^2_{(\sum_{x\in\mathcal{X}}\lambda_x\psi_d(x)\psi_d(x)^\top)^{-1}}/N. \tag{8.3}$$

Algorithm 24 GEMS-c Gap Elimination with Model Selection (Fixed Confidence)

Input: Number of iterations n, budget for dimension selection B and confidence parameter δ.

1: Set $\widehat{\mathcal{S}}_1 = \mathcal{Z}$.
2: **for** $k = 1, 2, \ldots, n$ **do**
3: Set $\delta_k = \delta/k^2$.
4: Define $g_k(d) := \max\{2^{2k}\,\iota(\mathcal{Y}(\psi_d(\widehat{\mathcal{S}}_k))), r_d(\zeta)\}$.
5: Get $d_k = \mathsf{OPT}(B, D, g_k(\cdot))$, where $d_k \leqslant D$ is largest dimension such that $g_k(d_k) \leqslant B$ (see Eq. (8.4) for the detailed optimization problem); set λ_k be the optimal design of the optimization problem $\inf_{\lambda\in\Delta_\mathcal{X}}\sup_{z,z'\in\widehat{\mathcal{S}}_k}\|\psi_{d_k}(z)-\psi_{d_k}(z')\|^2_{A_{d_k}(\lambda)^{-1}}$; set $N_k = \lceil g(d_k)2(1+\zeta)\log(|\widehat{\mathcal{S}}_k|^2/\delta_k)\rceil$.
6: Get allocation $\{x_1, \ldots, x_{N_k}\} = \mathsf{ROUND}(\lambda_k, N_k, d_k, \zeta)$.
7: Pull arms $\{x_1, \ldots, x_{N_k}\}$ and receive rewards $\{r_1, \ldots, r_{N_k}\}$.
8: Set $\widehat{\theta}_k = A_k^{-1}b_k \in \mathbb{R}^{d_k}$, where $A_k = \sum_{i=1}^{N_k}\psi_{d_k}(x_i)\psi_{d_k}(x_i)^\top$, and $b_k = \sum_{i=1}^{N_k}\psi_{d_k}(x_i)b_i$.
9: Set $\widehat{\mathcal{S}}_{k+1} = \widehat{\mathcal{S}}_k \setminus \{z \in \widehat{\mathcal{S}}_k : \exists z' \text{ s.t. } \langle\widehat{\theta}_k, \psi_{d_k}(z')-\psi_{d_k}(z)\rangle \geqslant \omega(z',z)\}$, where $\omega(z',z) := \|\psi_{d_k}(z')-\psi_{d_k}(z)\|_{A_k^{-1}}\sqrt{2\log(|\widehat{\mathcal{S}}_k|^2/\delta_k)}$.

Output: Set of uneliminated arms $\widehat{\mathcal{S}}_{n+1}$.

We now discuss the adaptive selection of hypothesis class, which is achieved through a new optimization problem: At round k, $d_k \in [D]$ is selected as the largest dimension such that the value of an experimental design is no larger than the fixed selection budget B, i.e.,

$$\begin{aligned} &\max d \\ &\text{s.t. } d \in [D], \end{aligned} \tag{8.4}$$

$$\max\left\{2^{2k}\cdot \inf_{\lambda\in\Delta_{\mathcal{X}}}\sup_{y\in\mathcal{Y}(\psi_d(\widehat{\mathcal{S}}_k))}\|y\|^2_{A_d(\lambda)^{-1}}, r_d(\zeta)\right\}\leqslant B.$$

The experimental design leverages the geometry of the *uneliminated* set of arms. Intuitively, the algorithm is selecting the *richest* hypothesis class that still allows the learner to improve its estimates of the gaps by a factor of 2 using (roughly) B samples. When the budget for dimension selection B is large enough, GEMS-c operates on well-specified linear bandits (i.e., using $d_k \geqslant d_\star$) at all rounds, guaranteeing that the output set of arms are (2^{1-n})-optimal. The next lemma provides guarantees for GEMS-c.

Lemma 8.7. *Suppose* $B \geqslant \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}$. *With probability at least* $1-\delta$, *GEMS-c outputs a set of arms* $\widehat{\mathcal{S}}_{n+1}$ *such that* $\Delta_z < 2^{1-n}$ *for any* $z \in \widehat{\mathcal{S}}_{n+1}$.

Algorithm 25 Adaptive Strategy for Model Selection (Fixed Confidence)

Input: Confidence parameter δ.

Randomly select a $\widehat{z}_\star \in \mathcal{Z}$ as the recommendation for the optimal arm.
for $\ell = 1, 2, \ldots$ **do**
  Set $\gamma_\ell = 2^\ell$ and $\delta_\ell = \delta/(2\ell^3)$.
  for $i = 1, 2, \ldots, \ell$ **do**
    Set $n_i = 2^i$, $B_i = \gamma_\ell / n_i = 2^{\ell - i}$, and get $\widehat{\mathcal{S}}_i = \text{GEMS-c}(n_i, B_i, \delta_\ell)$.
    if $\widehat{\mathcal{S}}_i = \{\widehat{z}\}$ is a singleton set **then**
      Update the recommendation $\widehat{z}_\star = \widehat{z}$.
      break (the inner for loop over i)

We present our main algorithm for model selection in Algorithm 25, which loops over an iterate ℓ with roughly geometrically increasing budget $\gamma_\ell = \ell 2^\ell$. Within each iteration ℓ, Algorithm 25 invokes GEMS-c ℓ times with different configurations (n_i, B_i): n_i is viewed as a guess for the unknown quantity $\log_2(1/\Delta_{\min})$; and B_i is viewed as a guess of $\rho^\star_{d_\star}$, which is then used to determine the adaptive selection hypothesis class. The configurations $\{(n_i, B_i)\}_{i=1}^{\ell}$ are chosen as the diagonal of a two dimensional gird over n_i and B_i. Within each iteration ℓ, the recommendation

$\widehat{z}_\star$ is updated as the arm contained in the *first* singleton set returned (if any). Since B_i is chosen in a decreasing order, we are recommending the arm selected from the richest hypothesis class that terminates recommending a single arm. The singleton is guaranteed to contain the optimal arm once a rich enough hypothesis class is considered. We provide the formal guarantees as follows.

Theorem 8.8. *Let* $\tau_\star = \log_2(4/\Delta_{\min}) \max\left\{\rho^\star_{d_\star}, r_{d_\star}(\zeta)\right\}$. *With probability at least* $1-\delta$, *Algorithm 25 starts to output the optimal arm within iteration* $\ell_\star = O(\log_2(\tau_\star))$, *and takes at most* $N = O(\tau_\star \log_2(\tau_\star) \log(|\mathcal{Z}| \log_2(\tau_\star)/\delta))$ *samples.*

The sample complexity in Theorem 8.8 is analyzed in an unverifiable way: Algorithm 25 starts to output the optimal arm after N samples, but it does not stop its sampling process. Nevertheless, up to a rounding-related term and other logarithmic factors,[3] the unverifiable sample complexity matches the non-interactive lower bound developed in Theorem 8.4. The non-interactive lower bound serves as a fairly strong baseline since the non-interactive learner is allowed to sample *with the knowledge of* $\theta_\star$. Computationally, Algorithm 25 starts to output the optimal arm after iteration $\ell_\star$, with at most $O(\ell_\star^2)$ subroutines (Algorithm 24) invoked. At each iteration $\ell \leqslant \ell_\star$, Algorithm 24 is invoked with configurations n_i, B_i such that $n_i B_i = 2^\ell \leqslant 2^{\ell_\star}$ (note that $\ell_\star$ is of logarithmic order). Up to a model selection step (i.e., selecting d_k), the per-round computational complexity of Algorithm 24 is similar to the complexity of the standard linear bandit algorithm RAGE.

Why not recommend arm verifiably. We provide a simple example to demonstrate that outputting the estimated best arm (using least squares) before examining full vectors in $\mathbb{R}^D$ can lead to incorrect answers, indicating that verifiable sample complexity, i.e., the number of samples required to terminate the game with a recommendation, scales with D $(\rho^\star_D)$. We consider a linear bandit problem with action

[3] We refer readers to Katz-Samuels and Jamieson (2020) for detailed discussion on unverifiable sample complexity. The rounding term $r_{d_\star}(\zeta) = O(d_\star/\zeta^2)$ commonly appears in the linear bandit pure exploration literature (Fiez et al., 2019; Katz-Samuels et al., 2020). Although we do not focus on optimizing logarithmic terms in this chapter, e.g., the $\log(|\mathcal{Z}|)$ term, our techniques can be extended to address this by combining techniques developed in Katz-Samuels et al. (2020).

set $\mathcal{X} = \mathcal{Z} = \{e_i\}_{i=1}^D$. We consider two cases: either (1) $\theta_\star := [1, 0, \ldots, 0, 0]^\top \in \mathbb{R}^D$ with $z_\star = e_1$; or (2) $\theta_\star := [1, 0, \ldots, 0, 2]^\top \in \mathbb{R}^D$ with $z_\star = e_D$. We assume *deterministic* feedback in this example. Let $n_x \geqslant 1$ denote the number of pulls on arm $x \in \mathcal{X}$. In both cases, for any $d < D$, the design matrix $\sum_{x \in \mathcal{X}} n_x \psi_d(x) \psi_d(x)^\top$ is diagonal with entries $(n_{e_i})_{i=1}^d$, and the least squares estimator is $\widehat{\theta}_d = e_1 \in \mathbb{R}^d$. As a result, e_1 will be recommended as the best arm: the recommendation is correct in the first case but incorrect in the second case. Essentially, one cannot rule out the possibility that $d_\star$ is equal to D without examining full vectors in $\mathbb{R}^D$. Verifiably identifying the best arm in $\mathbb{R}^D$ (with noisy feedback) takes $\widetilde{\Omega}(\rho^\star_D)$ samples (Fiez et al., 2019).

8.5 Fixed Budget Setting

We study the fixed budget setting with $\mathcal{Z} \subseteq \mathcal{X}$, which includes the linear bandit problem $\mathcal{Z} = \mathcal{X}$ as a special case. Similar to fixed confidence setting, we develop a main algorithm (Algorithm 27) that invokes a base algorithm as subroutines (GEMS-b, Algorithm 26). Algorithm 27 achieves an error probability $\widetilde{O}(\exp(-T/\rho^\star_{d_\star}))$, which, again, matches the strong baseline developed in Theorem 8.4.

The subroutine GEMS-b takes sample budget T, number of iterations n and dimension selection budget B as input, and outputs an (arbitrary) uneliminated arm after n iterations. As in the fixed confidence setting, GEMS-b performs adaptive selection of the hypothesis class through an optimization problem defined similar to the one in Eq. (8.4). The main differences from the fixed confidence subroutine is as follows: the selection budget B is only used for dimension selection, and the number of samples allocated per iteration is determined as $\lfloor T/n \rfloor$. GEMS-b is guaranteed to output the optimal arm with probability $1 - \widetilde{O}(\exp(-T/\rho^\star_{d_\star}))$ when the selection budget B is selected properly, as detailed in Lemma 8.9.

Lemma 8.9. *Suppose* $64\rho^\star_{d_\star} \leqslant B \leqslant 128\rho^\star_{d_\star}$ *and* $T/n \geqslant r_{d_\star}(\zeta) + 1$. *Algorithm 26 outputs an arm* $\widehat{z}_\star$ *such that* $\Delta_{\widehat{z}_\star} < 2^{1-n}$ *with probability at least*

$$1 - n|\mathcal{Z}|^2 \exp\left(-T/640\, n\, \rho^\star_{d_\star}\right).$$

Algorithm 26 GEMS-b Gap Elimination with Model Selection (Fixed Budget)

Input: Total budget T (allowing non-integer input), number of rounds n, budget for dimension selection B.

1: Set $T' = \lfloor T/n \rfloor$, $\widehat{\mathcal{S}}_1 = \mathcal{Z}$. Set $\widetilde{D}$ as the largest dimension that ensures rounding with T' samples, i.e., $\widetilde{D} = \mathsf{OPT}(T', D, f(\cdot))$, where $f(d) = r_d(\zeta)$.
2: **for** $k = 1, \dots, n$ **do**
3: Define function $g_k(d) := 2^{2k}\, \iota(\mathcal{Y}(\psi_d(\widehat{\mathcal{S}}_k)))$.
4: Get $d_k = \mathsf{OPT}(B, \widetilde{D}, g_k(\cdot))$, where where $d_k \leqslant \widetilde{D}$ is largest dimension such that $g_k(d_k) \leqslant B$ (similar to the optimization problem in Eq. (8.4)). Set λ_k be the optimal design of the optimization problem $\inf_{\lambda \in \Delta_{\mathcal{X}}} \sup_{z, z' \in \widehat{\mathcal{S}}_k} \|\psi_{d_k}(z) - \psi_{d_k}(z')\|^2_{A_{d_k}(\lambda)^{-1}}$.
5: Get allocations $\{x_1, \dots, x_{T'}\} = \mathsf{ROUND}(\lambda_k, T', d_k, \zeta)$.
6: Pull arms $\{x_1, \dots, x_{T'}\}$ and receive rewards $\{r_1, \dots, r_{T'}\}$.
7: Set $\widehat{\theta}_k = A_k^{-1} b_k \in \mathbb{R}^{d_k}$, where $A_k = \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\psi_{d_k}(x_i)^\top$, and $b_k = \sum_{i=1}^{N_k} \psi_{d_k}(x_i) b_i$.
8: Set $\widehat{\mathcal{S}}_{k+1} = \widehat{\mathcal{S}}_k \setminus \{z \in \widehat{\mathcal{S}}_k : \exists z' \text{ s.t. } \langle \widehat{\theta}_k, \psi_{d_k}(z') - \psi_{d_k}(z) \rangle \geqslant 2^{-k}\}$.

Output: Any uneliminated arm $\widehat{z}_\star \in \widehat{\mathcal{S}}_{n+1}$.

Our main algorithm for the fixed budget setting is introduced in Algorithm 27. Algorithm 27 consists of two phases: a pre-selection phase and a validation phase. The pre-selection phase collects a set of potentially optimal arms, selected by subroutines, and the validation phase examines the optimality of the collected arms. We provide Algorithm 27 with $2T$ total sample budget, and split the budget equally for each phase. At least one good subroutine is guaranteed to be invoked in the pre-selection phase (for sufficiently large T). The validation step focuses on identifying the best arm among the pre-selected $O((\log_2 T)^2)$ candidates (as explained in the next paragraph). Our selection-validation trick can be viewed as a *dimension-reduction* technique: we convert a linear bandit problem in $\mathbb{R}^D$ (with unknown $d_\star$) to another linear bandit problem in $\mathbb{R}^{O((\log_2 T)^2)}$,[4] i.e., a problem whose dimension

[4] Technically, we treat the problem as a standard multi-armed bandit problem with $O((\log_2 T)^2)$ arms, which is a special case of a linear bandit problem in $\mathbb{R}^{O((\log_2 T)^2)}$.

Algorithm 27 Adaptive Strategy for Model Selection (Fixed Budget)

Input: Total budget $2T$.

Step 1: Selection. Initialize an empty selection set $\mathcal{A} = \emptyset$.
Set $p = \lfloor W(T) \rfloor$ and $T' = T/p$.
for $i = 1, \ldots, p$ **do**
  Set $B_i = 2^i$, $q_i = \lfloor W(T'/B_i) \rfloor$ and $T'' = T'/q_i$.
  for $j = 1, \ldots, q_i$ **do**
    Set $n_j = 2^j$.
    Get $\widehat{z}_\star^{ij} = \text{GEMS-b}(T'', n_j, B_i)$ and insert $\widehat{z}_\star^{ij}$ into the pre-selection set $\mathcal{A}$.
Step 2: Validation. Pull each arm in the pre-selection set $\mathcal{A}$ exactly $\lfloor T/|\mathcal{A}| \rfloor$ times.

Output: Output arm $\widehat{z}_\star$ with the highest empirical reward from the validation step.

is only polylogarithmic in the budget T.

For non-negative variable p, we use $p = W(T)$ to represent the solution of equation $T = p \cdot 2^p$. One can see that $W(T) \leqslant \log_2 T$. As a result, at most $(\log_2 T)^2$ subroutines are invoked with different configurations of $\{(T'', n_j, B_i)\}$. The use of $W(\cdot)$ is to make sure that $T'' \geqslant n_j B_i$ for all subroutines invoked. This provides more efficient use of budget since the error probability upper bound guaranteed by GEMS-b scales as $\widetilde{O}(\exp(-T''/n_j B_i))$.

Theorem 8.10. *Suppose $\mathcal{Z} \subseteq \mathcal{X}$. If $T = \widetilde{\Omega}(\log_2(1/\Delta_{\min}) \max\{\rho^\star_{d_\star}, r_{d_\star}(\zeta)\})$, then Algorithm 27 outputs the optimal arm with error probability at most*

$$\log_2(4/\Delta_{\min})|\mathcal{Z}|^2 \exp\left(-\frac{T}{1024 \log_2(4/\Delta_{\min})\, \rho^\star_{d_\star}}\right) + 2(\log_2 T)^2 \exp\left(-\frac{T}{8(\log_2 T)^2/\Delta^2_{\min}}\right).$$

Furthermore, if there exist universal constants such that $\max_{x \in \mathcal{X}} \|\psi_{d_\star}(x)\|^2 \leqslant c_1$ and $\min_{z \in \mathcal{Z}} \|\psi_{d_\star}(z_\star) - \psi_{d_\star}(z)\|^2 \geqslant c_2$, the error probability is upper bounded by

$$O\left(\max\left\{ \log_2(1/\Delta_{\min})|\mathcal{Z}|^2, (\log_2 T)^2 \right\}\right.$$

$$\times \exp\left(-\frac{c_2 T}{\max\{\log_2(1/\Delta_{\min}), (\log_2 T)^2\} c_1 \rho^\star_{d_\star}}\right)\Bigg).$$

Under the mild assumption discussed above, the error probability of Algorithm 27 scales as $\widetilde{O}(\exp(-T/\rho^\star_{d_\star}))$. Such an error probability not only matches, up to logarithmic factors, the strong baseline developed in Theorem 8.4, but also matches the error bound in the non-model-selection setting (with known $d_\star$) (Katz-Samuels et al., 2020) (Algorithm 3 therein, which is also analyzed under a mild assumption). Computationally, Algorithm 27 invokes Algorithm 26 at most $(\log_2 T)^2$ times, each with budget $T'' \leqslant T$ and n_j, B_i such that $n_j B_i \leqslant T$. The per-round computational complexity of Algorithm 24 is similar to the one of Algorithm 26 (with similar configurations).

Compared to the fixed confidence setting, the fixed budget setting in linear bandits is relatively less studied (Hoffman et al., 2014; Katz-Samuels et al., 2020; Alieva et al., 2021; Yang and Tan, 2021). To our knowledge, even without the added challenge of model selection, near *instance optimal* error probability guarantee is only achieved by Algorithm 3 in Katz-Samuels et al. (2020). Our Algorithm 27 provides an alternative way to tackle the fixed budget setting, through a novel selection-validation procedure. Our techniques might be of independent interest.

8.6 Model Selection with Misspecification

We generalize the model selection problem into the *misspecified* regime in this section. Our goal here is to identify an ε-optimal arm due to misspecification. We aim to provide sample complexity/error probability guarantees with respect to a hypothesis class that is rich enough to allow us to identify an ε-optimal arm. Pure exploration with model misspecification are recently studied in the literature (Alieva et al., 2021; Camilleri et al., 2021; Zhu et al., 2021). The model selection criterion we consider here further complicates the problem setting and are not covered in previous work.

We consider the case where the expected reward $h(x)$ of any arm $x \in \mathcal{X} \cup$

$\mathcal{Z} \subseteq \mathbb{R}^D$ cannot be perfectly represented as a linear model in terms of its feature representation x. We use function $\widetilde{\gamma}(d)$ to capture the misspecification level with respect to truncation the level $d \in [D]$, i.e.,

$$\widetilde{\gamma}(d) := \min_{\theta \in \mathbb{R}^D} \max_{x \in \mathcal{X} \cup \mathcal{Z}} |h(x) - \langle \psi_d(\theta), \psi_d(x) \rangle|. \tag{8.5}$$

We use $\theta_\star^d \in \arg\min_{\theta \in \mathbb{R}^D} \max_{x \in \mathcal{X} \cup \mathcal{Z}} |h(x) - \langle \psi_d(\theta), \psi_d(x) \rangle|$ to denote (any) reward parameter that best captures the worst case deviation in $\mathbb{R}^d$, and use $\eta_d(x) := h(x) - \langle \psi_d(\theta_\star^d), \psi_d(x) \rangle$ to represent the corresponding misspecification with respect to arm $x \in \mathcal{X} \cup \mathcal{Z}$. We have $\max_{x \in \mathcal{X} \cup \mathcal{Z}} |\eta_d(x)| \leqslant \widetilde{\gamma}(d)$ by definition. Although the value of $\eta_d(x)$ depends on the selection of the possibly non-unique $\theta_\star^d$, only the worst-case deviation $\widetilde{\gamma}(d)$ is used in our analysis. Our results in this section are mainly developed in cases when $\mathcal{Z} \subseteq \mathcal{X}$, which contains the linear bandit problem $\mathcal{Z} = \mathcal{X}$ as a special case.

Proposition 8.11. *The misspecification level $\widetilde{\gamma}(d)$ is non-increasing with respect to d.*

The non-increasing property of $\widetilde{\gamma}(d)$ reflect the fact that the representation power of the linear component is getting better in higher dimensions. Following Zhu et al. (2021), we use $\gamma(d)$ to quantify the sub-optimality gap of the identified arm, i.e.,

$$\gamma(d) := \min \left\{ 2 \cdot 2^{-n} : n \in \mathbb{N}, \forall k \leqslant n, (2 + \sqrt{(1+\zeta)\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k)))})\widetilde{\gamma}(d) \leqslant 2^{-k}/2 \right\}.$$

It can be shown that, for any fixed $d \in [D]$, at least a $O(\sqrt{d}\,\widetilde{\gamma}(d))$-optimal arm can be identified in the existence of misspecification. Such inflation from $\widetilde{\gamma}(d)$ to $\sqrt{d}\,\widetilde{\gamma}(d)$ is unavoidable in general: Lattimore et al. (2020) constructs a hard instance such that identifying a $o(\sqrt{d}\widetilde{\gamma}(d))$-optimal arm requires sample complexity exponential in d, even with *deterministic* feedback. On the other hand, identifying a $\Omega(\sqrt{d}\,\widetilde{\gamma}(d))$-optimal arm only requires sample complexity polynomial in d. Such a sharp tradeoff between sample complexity and achievable optimality motivates our definition of $\gamma(d)$.

We assume $\gamma(d)$ can be made arbitrarily small for $d \in [D]$ large enough, which includes instances with no misspecification in $\mathbb{R}^D$ as special cases.[5] For any $\varepsilon > 0$, we define $d_\star(\varepsilon) := \min\{d \in [D] : \forall d' \geqslant d, \gamma(d') \leqslant \varepsilon\}$. We aim at identifying an ε-optimal arm with sample complexity related to $\rho^\star_{d_\star(\varepsilon)}$, which is defined as an ε-relaxed version of complexity measure $\rho^\star_{d_\star}$, i.e.,

$$\rho^\star_d(\varepsilon) := \inf_{\lambda \in \Delta_{\mathcal{X}}} \sup_{z \in \mathcal{Z} \setminus \{z_\star\}} \frac{\|\psi_d(z_\star) - \psi_d(z)\|^2_{A_d(\lambda)^{-1}}}{(\max\{h(z_\star) - h(z), \varepsilon\})^2}.$$

We consider a closely related complexity measure $\widetilde{\rho}^\star_d(\varepsilon)$, which is defined with respect to linear component $\widetilde{h}(x) := \langle \psi_d(\theta^d_\star), \psi_d(x) \rangle$, i.e.,

$$\widetilde{\rho}^\star_d(\varepsilon) := \inf_{\lambda \in \Delta_{\mathcal{X}}} \sup_{z \in \mathcal{Z} \setminus \{z_\star\}} \frac{\|\psi_d(z_\star) - \psi_d(z)\|^2_{A_d(\lambda)^{-1}}}{(\max\{\langle \psi_d(\theta^d_\star), \psi_d(z_\star) - \psi_d(z) \rangle, \varepsilon\})^2}.$$

Proposition 8.12 (Zhu et al. (2021)). *We have $\rho^\star_d(\varepsilon) \leqslant 9\widetilde{\rho}^\star_d(\varepsilon)$ for any $\varepsilon \geqslant \widetilde{\gamma}(d)$. Furthermore, if $\widetilde{\gamma}(d) < \Delta_{\min}/2$, $\widetilde{\rho}^\star_d(0)$ represents the complexity measure for best arm identification with respect to a linear bandit instance with action set $\mathcal{X}$, target set $\mathcal{Z}$ and reward function $\widetilde{h}(x) := \langle \psi_d(\theta^d_\star), \psi_d(x) \rangle$.*

Assuming $\widetilde{\gamma}(d_\star(\varepsilon)) < \min\{\varepsilon, \Delta_{\min}/2\}$, Proposition 8.12 shows that $\rho^\star_{d_\star(\varepsilon)}(\varepsilon)$ is at most a constant factor larger than $\widetilde{\rho}^\star_{d_\star(\varepsilon)}(\varepsilon)$, which is the ε-relaxed complexity measure of a closely related linear bandit problem (without misspecification) in $\mathbb{R}^{d_\star(\varepsilon)}$.

Fixed confidence setting. A modified algorithm (and its subroutine, both deferred to Section 8.9.5.2) is used for the fixed confidence setting with model misspecification. Sample complexity of the modified algorithm is provided as follows.

[5] We make this assumption in order to identify an ε-optimal arm for any pre-defined $\varepsilon > 0$. Otherwise, one can adjust the goal and identify arms with appropriate sub-optimality gaps.

Theorem 8.13. *With probability at least* $1-\delta$, *Algorithm 30 starts to output* 2ε*-optimal arms after* $N = \widetilde{O}(\log_2(1/\varepsilon)\max\{\rho^\star_{d_\star(\varepsilon)}(\varepsilon), r_{d_\star(\varepsilon)}(\zeta)\} + 1/\varepsilon^2)$ *samples, where we hide logarithmic terms besides* $\log_2(1/\varepsilon)$ *in the* $\widetilde{O}$ *notation.*

Remark 8.14. *The extra* $1/\varepsilon^2$ *term comes from a validation step in the modified algorithm. If the goal is to identify the optimal arm, then this term can be removed with a slight modification of the algorithm. See Section 8.9.5.3 for detailed discussion.*

Fixed budget setting. Our algorithms for the fixed budget setting are *robust* to model misspecification, and we provide the following guarantees.

Theorem 8.15. *Suppose* $\mathcal{Z} \subseteq \mathcal{X}$. *If* $T = \widetilde{\Omega}\left(\log_2(1/\varepsilon)\max\left\{\rho^\star_{d_\star(\varepsilon)}(\varepsilon), r_{d_\star(\varepsilon)}(\zeta)\right\}\right)$, *then Algorithm 27 outputs an* 2ε*-optimal arm with error probability at most*

$$\log_2(4/\varepsilon)|\mathcal{Z}|^2 \exp\left(-\frac{T}{4096\,\log_2(4/\varepsilon)\,\rho^\star_{d_\star(\varepsilon)}(\varepsilon)}\right) + 2(\log_2 T)^2 \exp\left(-\frac{T}{8(\log_2 T)^2/\varepsilon^2}\right).$$

Furthermore, if there exist universal constants such that $\max_{x\in\mathcal{X}}\|\psi_{d_\star(\varepsilon)}(x)\|^2 \leqslant c_1$ *and* $\min_{z\in\mathcal{Z}}\|\psi_{d_\star(\varepsilon)}(z_\star) - \psi_{d_\star(\varepsilon)}(z)\|^2 \geqslant c_2$, *the error probability is upper bounded by*

$$O\left(\max\left\{\log_2(1/\varepsilon)|\mathcal{Z}|^2, (\log_2 T)^2\right\} \times \exp\left(-\frac{c_2 T}{\max\left\{\log_2(1/\varepsilon), (\log_2 T)^2\right\} c_1 \rho^\star_{d_\star(\varepsilon)}(\varepsilon)}\right)\right).$$

8.7 Experiments

We empirically compare our Algorithm 25 with RAGE (Fiez et al., 2019), which shares a similar elimination structure to our subroutine (i.e., Algorithm 24) yet fails to conduct model selection in pure exploration. To our knowledge, besides

algorithms developed in this chapter, there is no other algorithm that can adapt to the model selection setup for pure exploration linear bandits.[6]

Problem instances. We conduct experiments with respect to the problem instance used to construct Proposition 8.6, which we detail as follows.

We consider a problem instance with $\mathcal{X} = \mathcal{Z} = \{x_i\}_{i=1}^{d_\star+1} \subseteq \mathbb{R}^{d_\star+1}$ such that $x_i = e_i$, for $i = 1, 2, \ldots, d_\star$ and $x_{d_\star+1} = (1-\varepsilon) \cdot e_{d_\star} + e_{d_\star+1}$, where e_i is the i-th canonical basis in $\mathbb{R}^{d_\star+1}$. The expected reward of each arm is set as $h(x_i) = \langle e_{d_\star}, x_i \rangle$, i.e., $\theta_\star = e_{d_\star}$. One can see that $d_\star$ is the intrinsic dimension and $D = d_\star + 1$ is the ambient dimension. We also notice that $x_\star = x_{d_\star}$ is the best arm with reward 1, $x_{d_\star+1}$ is the second best arm with reward $1-\varepsilon$ and all other arms have reward 0. The smallest sub-optimality gap is ε. We choose $d_\star = 9$, $D = 10$, and vary ε to control the instance-dependent complexity. By setting ε to be a small value, we create a problem instance such that $\rho_D^\star \gg \rho_{d_\star}^\star$: we have $\rho_{d_\star}^\star = O(d_\star)$ yet $\rho_D^\star = \Omega(1/\varepsilon^2)$ (see Section 8.9.2.4 for proofs).

Empirical evaluations. We evaluate the performance of each algorithm in terms of success rate, sample complexity and runtime. We conduct 100 independent trials for each algorithm. Both algorithms are force-stopped after reaching 10 million samples (denoted as the black line in Fig. 8.1). We consider an trial as failure if the algorithm fails to identify the best arm within 20 million samples. For each algorithm, we calculate the (unverifiable) sample complexity τ as the smallest integer such that the algorithm (1) empirically identifies the best arm; *and* (2) the algorithm won't change its recommendation for any later rounds $t > \tau$ (up to 20 million samples). The (empirical) runtime of the algorithm is calculated as the total time consumed up to round τ. We average sample complexities and runtimes with respect to succeeded trials.

[6] We defer additional experiment details/results to Section 8.9.6. The purpose of this section is to empirically demonstrate the importance of conducting model selection in pure exploration linear bandits, even on simple problem instances. We leave large-scale empirical evaluations for future work.

Table 8.1: Comparison of success rate with varying sub-optimality gap.

ε	10^{-2}	10^{-3}	10^{-4}	10^{-5}
RAGE	100%	98%	56%	62%
Ours	100%	100%	100%	100%

The success rates of RAGE and our algorithm are shown in Table 8.1. The success rate of RAGE drops dramatically as ε (the smallest sub-optimality gap) gets smaller. On the other hand, however, our algorithm is not affected by the change of ε since it automatically adapts to the intrinsic dimension $d_\star$: One can immediately see that $h(x_{d_\star}) \geqslant h(x_{d_\star+1})$ when working in $\mathbb{R}^{d_\star}$. Due to the same reason, our algorithm significantly outperforms RAGE in sample complexity as well (see Fig. 8.1): Our algorithm adapts to the true sample complexity $\rho^\star_{d_\star}$ yet RAGE suffers from complexity $\rho^\star_D \gg \rho^\star_{d_\star}$, especially when ε is small.

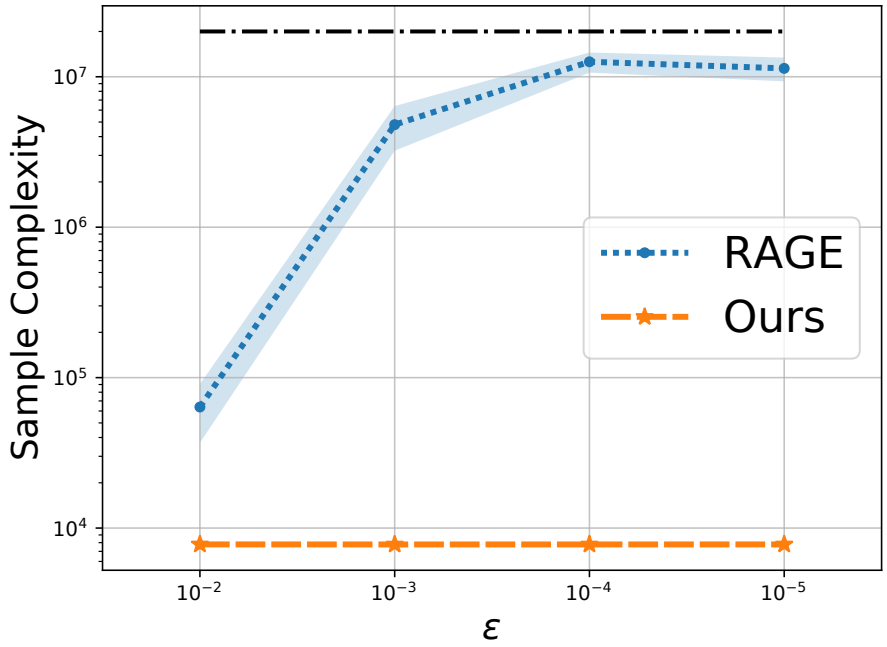


Figure 8.1: Comparison of sample complexity with varying sub-optimality gap.

The runtime of both algorithms are shown in Table 8.2. Our algorithm is affected by the computational overhead of conducting model selection (e.g., the two dimensional doubling trick). Thus, RAGE shows advantages in runtime when ε is relatively large. However, our algorithm runs faster than RAGE when ε gets smaller. This observation further shows that the implementation overhead can be small in comparison with the sample complexity gains achieved from model selection.

Table 8.2: Comparison of runtime with varying sub-optimality gap.

ε	10^{-2}	10^{-3}	10^{-4}	10^{-5}
RAGE	3.46 s	7.87 s	17.33 s	16.81 s
Ours	12.12 s	11.17 s	12.44 s	12.41 s

It is worth mentioning that simple variations of the problem instance studied in this section have long been considered as hard instances to examine linear bandit pure exploration algorithms (Soare et al., 2014; Xu et al., 2018; Tao et al., 2018; Fiez et al., 2019; Degenne et al., 2020). Our results show that, both theoretically and empirically, the problem instance becomes quite easy when viewed from the model selection perspective.

8.8 Discussion

We initiate the study of model selection in pure exploration linear bandits, in both fixed confidence and fixed budget settings, and design algorithms with near instance optimal guarantees. Along the way, we develop a novel selection-validation procedure to deal with the understudied fixed budget setting in linear bandits (even without the added challenge of model selection). We also adapt our algorithms to problems with model misspecification.

We conclude this chapter with some directions for future work. An immediate next step is to conduct large-scale evaluations for model selection in pure exploration linear bandits. One may need to develop practical version of our algorithms to bypass the computational overheads of conducting model selection. Another

interesting direction is provide guarantees to general transductive linear bandits, i.e., not restricted to cases $\mathcal{Z} \subseteq \mathcal{X}$, in fixed budget setting/misspecified regime. We believe one can use a selection-validation procedure similar to the one developed in Algorithm 26, but with the current validation step replaced by another linear bandit pure exploration algorithm. Note that the number of arms to be validated is of logarithmic order.

8.9 Proofs and Supporting Results

8.9.1 Supporting Results

8.9.1.1 Matrix Inversion and Rounding in Optimal Design

Our treatments are similar to the ones discussed in Zhu et al. (2021). We provide the details here for completeness.

Matrix Inversion. The notation $\|y\|^2_{A_d(\lambda)^{-1}}$ is clear when $A_d(\lambda)$ is invertible. For possibly singular $A_d(\lambda)$, pseudo-inverse is used if y belongs to the range of $A_{d(\lambda)}$; otherwise, we set $\|y\|^2_{A_d(\lambda)^{-1}} = \infty$. With this (slightly abused) definition of matrix inversion, we discuss how to do rounding next.

Rounding in Optimal Design. For any $\mathcal{S} \subseteq \mathcal{Z}$, the following optimal design

$$\inf_{\lambda \in \Delta_{\mathcal{X}}} \sup_{y \in \mathcal{Y}(\psi_d(\mathcal{S}))} \|y\|^2_{A_d(\lambda)^{-1}}$$

will select a design $\lambda^\star \in \Delta_{\mathcal{X}}$ such that every $y \in \mathcal{Y}(\psi_d(\mathcal{S}))$ lies in the range of $A_d(\lambda^\star)$.[7] If $\operatorname{span}(\mathcal{Y}(\psi_d(\mathcal{S}))) = \mathbb{R}^d$, then $A_d(\lambda^\star)$ is positive definite (recall that $A_d(\lambda^\star) = \sum_{x \in \mathcal{X}} \lambda_x \psi_d(x)\psi_d(x)^\top$ and $\operatorname{span}(\psi_d(\mathcal{X})) = \mathbb{R}^d$ comes from the assumption that $\operatorname{span}(\psi(\mathcal{X})) = \mathbb{R}^D$). Thus the rounding guarantees in Allen-Zhu et al. (2020) goes

[7] If the infimum is not attained, we can simply take a design $\lambda^{\star\star}$ with associated value $\tau^{\star\star} \leqslant (1+\zeta_0) \inf_{\lambda \in \Delta_{\mathcal{X}}} \sup_{y \in \mathcal{Y}(\psi_d(\mathcal{S}))} \|y\|^2_{A_d(\lambda)^{-1}}$ for a $\zeta_0 > 0$ arbitrarily small. This modification is used in our algorithms as well, and our results (bounds on sample complexity and error probability) goes through with changes only in constant terms.

through (Theorem 2.1 therein, which requires a positive definite design; with additional simple modifications dealt as in Appendix B of Fiez et al. (2019)).

We now consider the case when $A_d(\lambda^\star)$ is singular. Since $\text{span}(\psi_d(\mathcal{X})) = \mathbb{R}^d$, we can always find another λ' such that $A_d(\lambda')$ is invertible. For any $\zeta_1 > 0$, let $\widetilde{\lambda}^\star = (1-\zeta_1)\lambda^\star + \zeta_1\lambda'$. We know that $\widetilde{\lambda}^\star$ leads to a positive definite design. With respect to ζ_1, we can find another $\zeta_2 > 0$ small enough (e.g., smaller than the smallest eigenvalue of $\zeta_1 A_d(\lambda')$) such that $A_d(\widetilde{\lambda}^\star) \succeq A_d((1-\zeta_1)\lambda^\star) + \zeta_2 I$. Since $A_d((1-\zeta_1)\lambda^\star) + \zeta_2 I$ is positive definite, for any $y \in \mathcal{Y}(\psi_d(\mathcal{S}))$, we have

$$\|y\|^2_{A_d(\widetilde{\lambda}^\star)^{-1}} \leqslant \|y\|^2_{(A_d((1-\zeta_1)\lambda^\star)+\zeta_2 I)^{-1}}.$$

Fix any $y \in \mathcal{Y}(\psi_d(\mathcal{S}))$. Since y lies in the range of $A_d(\lambda^\star)$ (by definition of the objective and matrix inversion), we clearly have

$$\|y\|^2_{(A_d((1-\zeta_1)\lambda^\star)+\zeta_2 I)^{-1}} \leqslant \|y\|^2_{(A_d((1-\zeta_1)\lambda^\star))^{-1}} \leqslant \frac{1}{1-\zeta_1}\|y\|^2_{A_d(\lambda^\star)^{-1}}.$$

To summarize, we have

$$\|y\|^2_{A_d(\widetilde{\lambda}^\star)^{-1}} \leqslant \frac{1}{1-\zeta_1}\|y\|^2_{A_d(\lambda^\star)^{-1}},$$

where ζ_1 can be chosen arbitrarily small. We can thus send the positive definite design $\widetilde{\lambda}^\star$ to the rounding procedure in Allen-Zhu et al. (2020). We can incorporate the additional $1/(1-\zeta_1)$ overhead, for $\zeta_1 > 0$ chosen sufficiently small, into the sample complexity requirement $r_d(\zeta)$ of the rounding procedure.

8.9.1.2 Supporting Theorems and Lemmas

Lemma 8.16 ((Kaufmann et al., 2016)). *Fixed any pure exploration algorithm π. Let ν and ν' be two bandit instances with K arms such that the distribution ν_i and ν'_i are mutually absolutely continuous for all $i \in [K]$. For any almost-surely finite stopping time τ with respect to the filtration $\{\mathcal{F}_t\}_{t\geqslant 0}$, let $N_i(\tau)$ be the number of pulls on arm i at time τ.*

We then have

$$\sum_{i=1}^{K} \mathbb{E}_{\nu}[N_i(\tau)]\,\mathrm{KL}(\nu_i, \nu_i') \geqslant \sup_{\mathcal{E}\in\mathcal{F}_\tau} d(\mathbb{P}_{\nu}(\mathcal{E}), \mathbb{P}_{\nu'}(\mathcal{E})),$$

where $d(x,y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$ *for* $x, y \in [0,1]$ *and with the convention that* $d(0,0) = d(1,1) = 0$.

The following two lemmas largely follow the analysis in Fiez et al. (2019).

Lemma 8.17. *Let* $\mathcal{S}_k = \{z \in \mathcal{Z} : \Delta_z < 4\cdot 2^{-k}\}$. *We then have*

$$\sup_{k\in[\lfloor \log_2(4/\varepsilon)\rfloor]} \left\{2^{2k}\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k)))\right\} \leqslant 64\rho_d^\star(\varepsilon), \tag{8.6}$$

and

$$\sup_{k\in[\lfloor \log_2(4/\varepsilon)\rfloor]} \left\{\max\left\{2^{2k}\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k))), r_d(\zeta)\right\}\right\} \leqslant \max\{64\rho_d^\star(\varepsilon), r_d(\zeta)\}, \tag{8.7}$$

where ζ *is the rounding parameter.*

Proof. For $y = \psi_d(z_\star) - \psi_d(z)$, we define $\Delta_y = \Delta_z = h(z_\star) - h(z)$. We have that

$$\begin{aligned}
\rho_d^\star(\varepsilon) &= \inf_{\lambda\in\Delta_x} \sup_{y\in\mathcal{Y}^\star(\psi_d(\mathcal{Z}))} \frac{\|y\|^2_{A_d(\lambda)^{-1}}}{\max\{\Delta_y, \varepsilon\}^2} \\
&= \inf_{\lambda\in\Delta_x} \sup_{k\in[\lfloor \log_2(4/\varepsilon)\rfloor]} \sup_{y\in\mathcal{Y}^\star(\psi_d(\mathcal{S}_k))} \frac{\|y\|^2_{A_d(\lambda)^{-1}}}{\max\{\Delta_y, \varepsilon\}^2} \\
&\geqslant \sup_{k\in[\lfloor \log_2(4/\varepsilon)\rfloor]} \inf_{\lambda\in\Delta_x} \sup_{y\in\mathcal{Y}^\star(\psi_d(\mathcal{S}_k))} \frac{\|y\|^2_{A_d(\lambda)^{-1}}}{\max\{\Delta_y, \varepsilon\}^2} \\
&> \sup_{k\in[\lfloor \log_2(4/\varepsilon)\rfloor]} \inf_{\lambda\in\Delta_x} \sup_{y\in\mathcal{Y}^\star(\psi_d(\mathcal{S}_k))} \frac{\|y\|^2_{A_d(\lambda)^{-1}}}{(4\cdot 2^{-k})^2} \qquad (8.8) \\
&\geqslant \sup_{k\in[\lfloor \log_2(4/\varepsilon)\rfloor]} \inf_{\lambda\in\Delta_x} \sup_{y\in\mathcal{Y}(\psi_d(\mathcal{S}_k))} \frac{\|y\|^2_{A_d(\lambda)^{-1}}/4}{(4\cdot 2^{-k})^2} \qquad (8.9)
\end{aligned}$$

$$\geqslant \sup_{k\in[\lfloor \log_2(4/\varepsilon)\rfloor]} 2^{2k}\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k)))/64,$$

where Eq. (8.8) comes from the fact that $4\cdot 2^{-k}\geqslant\varepsilon$ when $k\leqslant\lfloor\log_2(4/\varepsilon)\rfloor$; Eq. (8.9) comes from the fact that $\psi_d(z)-\psi_d(z')=(\psi_d(z)-\psi_d(z_\star))+(\psi_d(z_\star)-\psi_d(z'))$. This implies that, for any $k\in[\lfloor\log_2(4/\varepsilon)\rfloor]$,

$$\max\{2^{2k}\rho(\mathcal{Y}(\psi_d(\mathcal{S}_k))), r_d(\zeta)\}\leqslant\max\{64\rho_d^\star(\varepsilon), r_d(\zeta)\}.$$

And the desired Eq. (8.7) immediately follows. □

Lemma 8.18. *Let* $\mathcal{S}_k=\{z\in\mathcal{Z}:\Delta_z<4\cdot 2^{-k}\}$. *We then have*

$$\sup_{k\in[\lceil\log_2(4/\Delta_{\min})\rceil]}\{2^{2k}\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k)))\}\leqslant 64\rho_d^\star, \tag{8.10}$$

and

$$\sup_{k\in[\lceil\log_2(4/\Delta_{\min})\rceil]}\{\max\{2^{2k}\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k))), r_d(\zeta)\}\}\leqslant\max\{64\rho_d^\star, r_d(\zeta)\}, \tag{8.11}$$

where ζ *is the rounding parameter.*

Proof. Take $\varepsilon=\Delta_{\min}$ in Lemma 8.17. □

The following lemma largely follows the analysis in Soare et al. (2014), with generalization to the transductive setting and more careful analysis in terms of matrix inversion.

Lemma 8.19. *Fix* $\mathcal{Z}\subseteq\mathcal{X}\subseteq\mathbb{R}^D$. *Suppose* $\max_{x\in\mathcal{X}}\|x\|^2\leqslant c_1$ *and* $\min_{z\in\mathcal{Z}\setminus\{z_\star\}}\|z_\star-z\|^2\geqslant c_2$ *with some absolute constant* c_1 *and* c_2. *We have*

$$\frac{c_2}{c_1\Delta_{\min}^2}\leqslant\rho^\star:=\inf_{\lambda\in\Delta_{\mathcal{X}}}\sup_{z\in\mathcal{Z}\setminus\{z_\star\}}\frac{\|z_\star-z\|^2_{A(\lambda)^{-1}}}{\Delta_z^2},$$

where $\Delta_{\min}=\min_{z\in\mathcal{Z}\setminus\{z_\star\}}\{\Delta_z\}$.

Proof. Let $\lambda^\star$ be the optimal design that attains $\rho^\star$;[8] and let $z' \in \mathcal{Z}$ be any arm with the smallest sub-optimality gap $\Delta_{\min}$. We then have

$$\begin{aligned}\rho^\star &= \max_{z \in \mathcal{Z} \setminus \{z_\star\}} \frac{\|z_\star - z\|^2_{A(\lambda^\star)^{-1}}}{\Delta_z^2} \\ &\geqslant \frac{\|z_\star - z'\|^2_{A(\lambda^\star)^{-1}}}{\Delta_{z'}^2} \\ &= \frac{\|z_\star - z'\|^2_{A(\lambda^\star)^{-1}}}{\Delta_{\min}^2},\end{aligned} \tag{8.12}$$

where $z_\star - z'$ necessarily lie in the range of $A(\lambda^\star)$ according to the definition of matrix inversion in Section 8.9.1.1.

We now lower bound $\|z_\star - z'\|^2_{A(\lambda^\star)^{-1}}$. Note that $A(\lambda^\star)$ is positive semi-definite. We write $A(\lambda^\star) = Q\Sigma Q^\top$ where Q is an orthogonal matrix and Σ is a diagonal matrix storing eigenvalues. We assume that the last k eigenvalues of Σ are zero. Let $\gamma_{\max} = \|A(\lambda^\star)\|_2 = \|\Sigma\|_2$ be the largest eigenvalue, we have $\gamma_{\max} \leqslant \max_{x \in \mathcal{X}} \|x\|^2 \leqslant c_1$ since $A(\lambda^\star) = \sum_{x \in \mathcal{X}} \lambda^\star(x) x x^\top$ and $\sum_{x \in \mathcal{X}} \lambda^\star(x) = 1$. Let $w = Q^\top(z_\star - z')$. Since $z_\star - z'$ is in the range of $A(\lambda^\star)$, we know that the last k entries of w must be zero. We then have

$$\begin{aligned}\|z_\star - z'\|^2_{A(\lambda^\star)^{-1}} &= (z_\star - z)^\top A(\lambda^\star)^{-1}(z_\star - z) \\ &= w^\top \Sigma^{-1} w \\ &\geqslant \|w\|^2 / c_1 \\ &\geqslant c_2 / c_1,\end{aligned} \tag{8.13}$$

where Eq. (8.13) comes from fact that $\|w\|^2 = \|z_\star - z'\|^2$ and the assumption $\|z_\star - z\|^2 \geqslant c_2$ for all $z \in \mathcal{Z}$. □

Lemma 8.20. *The following statements hold.*

1. $T \geqslant 4a \log 2a \implies T \geqslant a \log_2 T$ *for* $T, a > 0$.

[8] If the infimum is not attained, one can apply the argument that follows with a limit sequence. See footnote in Section 8.9.1.1 for more details on how to construct an approximating design.

2. $T \geqslant 16a\,(\log 16a)^2 \implies T \geqslant a\,(\log_2 T)^2$ *for* $T, a > 1$.

Proof. We first recall that $T \geqslant 2a\log a \implies T \geqslant a\log T$ for $T, a > 0$ (Shalev-Shwartz and Ben-David, 2014). Since $\log_2 T = \log T/\log 2 < 2\log T$, the first statement immediately follows.

To prove the second statement, we only need to find conditions on T such that $T \geqslant 4a\,(\log T)^2$. Note that we have $\sqrt{T} \geqslant 8\sqrt{a}\log 4\sqrt{a} = 4\sqrt{a}\log 16a \implies \sqrt{T} \geqslant 4\sqrt{a}\log\sqrt{T} = 2\sqrt{a}\log T$. For $T, a > 1$, this is equivalent to $T \geqslant 16a\,(\log 16a)^2 \implies T \geqslant 4a\,(\log T)^2 \geqslant a\,(\log_2 T)^2$, and thus the second statement follows. □

8.9.1.3 Supporting Algorithms

Algorithm 28 OPT

Input: Selection budget B, dimension upper bound D and selection function $g(\cdot)$ (which is a function of the dimension $d \in [D]$).

1: Get d_k such that

$$\begin{aligned} d_k = &\max d \\ &\text{s.t. } g(d) \leqslant B, \text{ and } d \in [D]. \end{aligned}$$

Output: The selected dimension d_k.

8.9.2 Proofs and Supporting Results for Section 8.3

8.9.2.1 Proof of Theorem 8.3

Theorem 8.3. *Suppose* $\xi_t \sim \mathcal{N}(0,1)$ *for all* $t \in \mathbb{N}_+$ *and* $\delta \in (0, 0.15]$. *Any* δ*-PAC algorithm with respect to* $(\mathcal{X}, \mathcal{Z}, \theta_\star \in \Theta_{d_\star})$ *with stopping time* τ *satisfies* $\mathbb{E}_{\theta_\star}[\tau] \geqslant \rho^\star_{d_\star}\log(1/2.4\delta)$.

Proof. The proof of the theorem mostly follows the proof of lower bound in Fiez et al. (2019). We additionally consider the model selection problem $(\mathcal{X}, \mathcal{Z}, \theta_\star \in \Theta_{d_\star})$ and carefully deal with the matrix inversion.

Consider the instance $(\mathcal{X}, \mathcal{Z}, \theta_\star \in \Theta_{d_\star})$, where $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\text{span}(\mathcal{X}) = \mathbb{R}^D$, $\mathcal{Z} = \{z_1, \ldots, z_m\}$. Suppose that $z_1 = \arg\max_{z \in \mathcal{Z}} \langle \theta_\star, z \rangle$. We consider the alternative set $\mathcal{C}_{d_\star} := \{\theta \in \Theta_{d_\star} : \exists i \in [m] \text{ s.t. } \langle \theta, z_1 - z_i \rangle < 0\}$, where z_1 is not the best arm for any $\theta \in \mathcal{C}_{d_\star}$. Following the "change of measure" argument in Lemma 8.16, we know that $\mathbb{E}_{\theta_\star}[\tau] \geqslant \tau^\star$, where $\tau^\star$ is the solution of the following constrained optimization

$$\tau^\star := \min_{t_1, \ldots, t_n \in \mathbb{R}_+} \sum_{i=1}^n t_i \tag{8.14}$$
$$\text{s.t.} \inf_{\theta \in \mathcal{C}_{d_\star}} \sum_{i=1}^n t_i \, \text{KL}(\nu_{\theta_\star, i}, \nu_{\theta, i}) \geqslant \log(1/2.4\delta),$$

where we use the notation $\nu_{\theta,i} = \mathcal{N}(\langle \theta, x_i \rangle, 1) = \mathcal{N}(\langle \psi_{d_\star}(\theta), \psi_{d_\star}(x_i) \rangle, 1)$ (due to the fact that $\theta \in \mathcal{C}_{d_\star}$). We also have $\text{KL}(\nu_{\theta_\star, i}, \nu_{\theta, i}) = \frac{1}{2} \langle \psi_{d_\star}(\theta_\star) - \psi_{d_\star}(\theta), \psi_{d_\star}(x_i) \rangle^2$.

We next show that for any $t = (t_1, \ldots, t_n)^\top \in \mathbb{R}_+^n$ satisfies the constraint of Eq. (8.14), we must have $\psi_{d_\star}(z_1) - \psi_{d_\star}(z_i) \in \text{span}(\{\psi_{d_\star}(x_i) : t_i > 0\}), \forall 2 \leqslant i \leqslant m$. Suppose not, there must exists a $\psi_{d_\star}(u) \in \mathbb{R}^{d_\star}$ such that (1) $\langle \psi_{d_\star}(u), \psi_{d_\star}(x_i) \rangle = 0$ for all $i \in [n]$ such that $t_i > 0$; and (2) there exists a $2 \leqslant j \leqslant m$ such that $\langle \psi_{d_\star}(z_1) - \psi_{d_\star}(z_j), \psi_{d_\star}(u) \rangle \neq 0$. Suppose $\langle \psi_{d_\star}(z_1) - \psi_{d_\star}(z_j), \psi_{d_\star}(u) \rangle > 0$ (the other direction is similar), we can choose a $\theta' \in \Theta_{d_\star}$ such that the first $d_\star$ coordinates of θ' equals to $\psi_{d_\star}(\theta_\star) - \alpha \psi_{d_\star}(u)$ for a $\alpha > 0$ large enough (so that $\theta' \in \mathcal{C}_{d_\star}$). With such θ', however, we have

$$\sum_{i=1}^n t_i \, \text{KL}(\nu_{\theta_\star, i}, \nu_{\theta', i}) = \sum_{i=1}^n t_i \frac{1}{2} \langle \alpha \psi_{d_\star}(u), \psi_{d_\star}(x_i) \rangle^2 = 0 < \log(1/2.4\delta),$$

which leads to a contradiction. As a result, we can safely calculate $\|\psi_{d_\star}(z_1) - \psi_{d_\star}(z_i)\|^2_{A_{d_\star}(t)^{-1}}$ or $A_{d_\star}(t)^{-1}(\psi_{d_\star}(z_1) - \psi_{d_\star}(z_i))$ where $A_{d_\star}(t) := \sum_{i=1}^n t_i \psi_{d_\star}(x_i) \psi_{d_\star}(x_i)^\top / \bar{t}$ and $\bar{t} := \sum_{i=1}^n t_i$. The rest of the proof follows from the proof of theorem 1 in Fiez et al. (2019). □

8.9.2.2 Proof of Theorem 8.4

Theorem 8.4. *Fix* $(\mathcal{X}, \mathcal{Z}, \theta_\star \in \Theta_{d_\star})$ *and* $\delta \in (0, 0.015]$. *Any non-interactive algorithm* $\mathscr{A}$ *using a feature mappings of dimension* $d \geqslant d_\star$ *makes a mistake with probability at least* δ *as long as it uses no more than* $\frac{1}{2}\rho^\star_{d_\star} \log(1/\delta)$ *samples.*

Proof. The proof largely follows from the proof of Theorem 3 in Katz-Samuels et al. (2020) (but ignore the $\gamma^\star$ term therein. We are effectively using a weaker lower bound, yet it suffices for our purpose.). The non-interactive MLE uses at least $\frac{1}{2}\rho^\star_d \log(1/\delta)$ with respect to any feature mapping $\psi_d(\cdot)$ for $d_\star \leqslant d \leqslant D$. The statement then follows from the monotonicity of $\{\rho^\star_d\}_{d=d_\star}^D$ as shown in Proposition 8.5. □

8.9.2.3 Proof of Proposition 8.5

Proposition 8.5. *The monotonic relation* $\rho^\star_{d_1} \leqslant \rho^\star_{d_2}$ *holds true for any* $d_\star \leqslant d_1 \leqslant d_2 \leqslant D$.

Proof. We first prove equivalence results in the general setting in Step 1, 2 and 3; and then apply the results to the model selection problem in Step 4 to prove monotonicity over $\{\rho^\star_d\}_{d=d_\star}^D$.

We consider instance $(\mathcal{X}, \mathcal{Z}, \theta_\star)$ in the general setting, where $\mathcal{X} = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$, $\text{span}(\mathcal{X}) = \mathbb{R}^d$, $\mathcal{Z} = \{z_1, \ldots, z_m\}$ and $\theta_\star \in \mathbb{R}^d$. We suppose that $z_1 = \arg\max_{z \in \mathcal{Z}} \langle \theta_\star, z \rangle$ is the unique optimal arm and $\text{span}(\{z_1 - z\}_{z \in \mathcal{Z} \setminus \{z_1\}}) = \mathbb{R}^d$. We use the notations $y_j := z_1 - z_j$ for $j = 2, \ldots, m$, and $\nu_{\theta,i} := \mathcal{N}(x_i^\top \theta, 1)$. For any $t = (t_1, \ldots, t_n)^\top \in \mathbb{R}^n_+$, we also use the notation $A(t) = \sum_{i=1}^n t_i x_i x_i^\top \in \mathbb{R}^{d \times d}$ to denote a design matrix with respect to t (t doesn't need to be inside the simplex $\Delta_\mathcal{X}$). We consider any fixed $\delta \in (0, 0.15]$.

Step 1: Closure of constraints. Let $\mathcal{C}$ denote the set of parameters where z_1 is no longer the best arm anymore, i.e.,

$$\mathcal{C} := \{\theta \in \mathbb{R}^d : \exists i \in [m] \text{ s.t. } \theta^\top (z_1 - z_i) < 0\}.$$

Using the "change of measure" argument from Kaufmann et al. (2016), the lower bound is given by the following optimization problem (Audibert et al., 2010; Fiez

et al., 2019)

$$\tau^\star := \min_{t_1,\dots,t_n \in \mathbb{R}_+} \sum_{i=1}^n t_i$$
$$\text{s.t. } \inf_{\theta \in \mathcal{C}} \sum_{i=1}^n t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) \geqslant \log(1/2.4\delta).$$

First, we show that the value $\tau^\star$ equals to the value of another optimization problem, i.e.,

$$\tau^\star = \min_{t_1,\dots,t_n \in \mathbb{R}_+} \sum_{i=1}^n t_i$$
$$\text{s.t. } \min_{\theta \in \bar{\mathcal{C}}} \sum_{i=1}^n t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) \geqslant \log(1/2.4\delta),$$

where $\bar{\mathcal{C}} = \{\theta \in \mathbb{R}^d : \exists i \in [m] \text{ s.t. } \theta^\top (z_1 - z_i) \leqslant 0\}$. Note that that we must show that the minimum in the constraint is attained, i.e., the $\min_{\theta \in \bar{\mathcal{C}}}$ part. We first show the equivalence between the original problem and the problem with respect to $\inf_{\theta \in \bar{\mathcal{C}}}$; and then show the equivalence between problems with respect to $\inf_{\theta \in \bar{\mathcal{C}}}$ and $\min_{\theta \in \bar{\mathcal{C}}}$. We fix any $t = (t_1, \dots, t_n)^\top \in \mathbb{R}_+^n$.

Step 1.1: We claim that $\inf_{\theta \in \mathcal{C}} \sum_{i=1}^n t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) \geqslant \log(1/2.4\delta)$ if and only if $\inf_{\theta \in \bar{\mathcal{C}}} \sum_{i=1}^n t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) \geqslant \log(1/2.4\delta)$.

Since $\bar{\mathcal{C}} \supset \mathcal{C}$, the $\Longleftarrow$ direction is obvious.

Now, suppose $\inf_{\theta \in \bar{\mathcal{C}}} \sum_{i=1}^n t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) < \log(1/2.4\delta)$. By definition of inf, there exists $\theta_0 \in \bar{\mathcal{C}}$ such that

$$\sum_{i=1}^n t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta_0,i}) < \log(1/2.4\delta).$$

Since $\bar{\mathcal{C}}$ is the closure of an open set $\mathcal{C}$, there exists a sequence $\{\theta_j\}$ in $\mathcal{C}$ approaching

θ_0. Note that

$$\sum_{i=1}^{n} t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) = \sum_{i=1}^{n} t_i \frac{1}{2} (x_i^\top (\theta_\star - \theta))^2 = \frac{1}{2} \|\theta_\star - \theta\|^2_{A(t)}.$$

Then, by the continuity of $\frac{1}{2}\|\theta_\star - \theta\|^2_{A(t)}$ in θ, there exists a $\theta \in \mathcal{C}$ such that $\sum_{i=1}^{n} t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) < \log(1/2.4\delta)$. This gives a contradiction and thus proves the $\Longrightarrow$ direction.

Step 1.2: Now, we must show that the infimum is attained whenever

$$\inf_{\theta \in \bar{\mathcal{C}}} \sum_{i=1}^{n} t_i \, \mathrm{KL}(\nu_{\theta_\star,i} \| \nu_{\theta,i}) \geqslant \log(1/2.4\delta),$$

that is, there exists $\theta_0 \in \bar{\mathcal{C}}$ such that

$$\sum_{i=1}^{n} t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta_0,i}) = \inf_{\theta \in \bar{\mathcal{C}}} \sum_{i=1}^{n} t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}).$$

Claim: Fix $t = (t_1, \ldots, t_n)^\top \in \mathbb{R}^n_+$. If $\mathrm{span}(\{x_i : t_i > 0\}) \neq \mathbb{R}^d$, then

$$\inf_{\theta \in \bar{\mathcal{C}}} \sum_{i=1}^{n} t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) < \log(1/2.4\delta).$$

First, we show the claim. Fix $t = (t_1, \ldots, t_n)^\top \in \mathbb{R}^n_+$ and suppose $\mathrm{span}(\{x_i : t_i > 0\}) \neq \mathbb{R}^d$. Since $\mathrm{span}(\{x_i : t_i > 0\}) \neq \mathbb{R}^d$, there exists $u \in \mathbb{R}^d$ such that $u^\top x_i = 0$ for all i such that $t_i > 0$. Since $\{z_1 - z_i : i \in [m]\}$ spans $\mathbb{R}^d$ by assumption, there exists $i \in [m]$ such that $u^\top (z_1 - z_i) \neq 0$. Suppose that $u^\top (z_1 - z_i) < 0$ (the other case is similar). Then, there exists a sufficiently large $\alpha > 0$ such that $(\theta_\star + \alpha u)^\top (z_1 - z_i) < 0$, implying that $\theta_\star + \alpha u \in \mathcal{C}$. Moreover, by construction of u, we have

$$\sum_{i=1}^{n} t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta_\star+\alpha u,i}) = \sum_{i=1}^{n} t_i \frac{1}{2} (x_i^\top (\alpha u))^2 = \sum_{i : t_i > 0} t_i \frac{1}{2} (x_i^\top (\alpha u))^2 = 0 < \log(1/2.4\delta),$$

and thus leads to the claim.

Now, suppose $\inf_{\theta\in\bar{\mathcal{C}}} \sum_{i=1}^n t_i \,\mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) \geqslant \log(1/2.4\delta)$. Then, $\mathrm{span}(\{x_i : t_i > 0\}) = \mathbb{R}^d$. Then, $\|\cdot\|_{A(t)}^2$ is a norm, and the set

$$\left\{\theta \in \mathbb{R}^d : \frac{1}{2}\|\theta - \theta_\star\|_{A(t)}^2 \leqslant \varepsilon\right\}$$

is compact for every ε. Then, since $\bar{\mathcal{C}}$ is closed and $\frac{1}{2}\|\theta - \theta_\star\|_{A(t)}^2$ has compact sublevel sets, there exists a $\theta_0 \in \bar{\mathcal{C}}$ such that

$$\sum_{i=1}^n t_i \,\mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta_0,i}) = \inf_{\theta\in\bar{\mathcal{C}}} \sum_{i=1}^n t_i \,\mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}).$$

This shows the equivalence between problems with respect to $\inf_{\theta\in\bar{\mathcal{C}}}$ and $\min_{\theta\in\bar{\mathcal{C}}}$.

Step 2: Rewrite the optimization problem. Define

$$\bar{\mathcal{C}}_i = \{\theta \in \mathbb{R}^d : \theta^\top (z_1 - z_i) \leqslant 0\},$$

and note that $\bar{\mathcal{C}} = \cup_{i=1}^m \bar{\mathcal{C}}_i$. Observe that

$$\begin{aligned}
\tau^\star &:= \min_{t_1,\dots,t_n\in\mathbb{R}_+} \sum_{i=1}^n t_i \\
&\qquad \text{s.t. } \min_{\theta\in\bar{\mathcal{C}}} \sum_{i=1}^n t_i \,\mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) \geqslant \log(1/2.4\delta) \\
&= \min_{t_1,\dots,t_n\in\mathbb{R}_+} \sum_{i=1}^n t_i \\
&\qquad \text{s.t. } \min_{i\in[m]} \min_{\theta\in\bar{\mathcal{C}}_i} \sum_{i=1}^n t_i \,\mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) \geqslant \log(1/2.4\delta).
\end{aligned}$$

Consider the optimization problem:

$$\min_{\theta\in\bar{\mathcal{C}}_i} \frac{1}{2} \sum_{i=1}^n t_i (x_i^\top (\theta_\star - \theta))^2 = \min_{\theta\in\bar{\mathcal{C}}_i} \frac{1}{2}\|\theta_\star - \theta\|_{A(t)}^2$$

Note that since the objective is convex and there exists $\theta \in \mathbb{R}^d$ such that $\theta^\top(z_1 - z_i) < 0$, Slater's condition holds and, therefore, strong duality holds. We form the Lagrangian with lagrange multiplier $\gamma \in \mathbb{R}_+$ to obtain

$$\text{Ł}(\theta, \gamma) = \frac{1}{2}\|\theta_\star - \theta\|^2_{A(t)} + \gamma \cdot y_i^\top \theta$$

Differentiating with respect to θ and γ, we have that (note that $A(t)$ is invertible from the claim in Step 1)

$$\begin{cases} \theta & = \theta_\star - \gamma A(t)^{-1} y_i, \\ y_i^\top \theta & = 0. \end{cases}$$

These imply that $\theta_0 := \theta_\star - \frac{y_i^\top \theta_\star A(t)^{-1} y_i}{y_i^\top A(t)^{-1} y_i}$ and $\gamma_0 := \frac{y_i \top \theta_\star}{y_i^\top A(t)^{-1} y_i} \in \mathbb{R}_+$ satisfy the K.K.T. conditions, and $\theta = \theta_0$ is the minimizer (primal optimal solution) of the constrained optimization problem (note that it's a convex program). Therefore, we have

$$\min_{\theta \in \tilde{\mathcal{C}}_i} \frac{1}{2} \sum_{i=1}^n t_i (x_i^\top(\theta_\star - \theta))^2 = \frac{(y_i^\top \theta_\star)^2}{\|y_i\|^2_{A(t)^{-1}}}$$

In conclusion, we have

$$\begin{aligned} \tau^\star = \min_{t_1,\dots,t_n \in \mathbb{R}_+} & \sum_{i=1}^n t_i \\ \text{s.t. } & \frac{(y_j^\top \theta_\star)^2}{\|y_j\|^2_{A(t)^{-1}}} \geqslant \log(1/2.4\delta), \forall\, 2 \leqslant j \leqslant m. \end{aligned}$$

Step 3: Re-express the optimization problem. Furthermore, we have that

$$\begin{aligned} \tau^\star = \min_{s, t_1,\dots,t_n \in \mathbb{R}_+} & s \\ \text{s.t. } & (y_j^\top \theta_\star)^2 \geqslant \log(1/2.4\delta)\|y_j\|^2_{A(t)^{-1}}, \forall\, 2 \leqslant j \leqslant m \end{aligned} \tag{8.15}$$

$$s \geqslant \sum_{i=1}^{n} t_i.$$

Rearranging these constraints, we have that

$$s \geqslant \sum_{i=1}^{n} t_i \geqslant \log(1/2.4\delta) \sum_{i=1}^{n} t_i \frac{\|y_j\|^2_{A(t)^{-1}}}{(y_j^\top \theta_\star)^2} = \log(1/2.4\delta) \frac{\|y_j\|^2_{A(\lambda)^{-1}}}{(y_j^\top \theta_\star)^2}, \forall\, 2 \leqslant j \leqslant m.$$

We do a change of variables $\lambda \in \Delta_\mathcal{X}$ and $\lambda_i = \frac{t_i}{\sum_{i=1}^n t_i}$, and the optimization problem is equivalent to

$$\begin{aligned} \tau^\star = \min_{s \in \mathbb{R}_+, \lambda \in \Delta_\mathcal{X}} & \; s \\ \text{s.t. } & s \geqslant \max_{j=2,\dots,m} \log(1/2.4\delta) \frac{\|y_j\|^2_{A(\lambda)^{-1}}}{(y_j^\top \theta_\star)^2}. \end{aligned}$$

Thus, we have that

$$\tau^\star \geqslant \inf_{\lambda \in \Delta_\mathcal{X}} \max_{j=2,\dots,m} \frac{\|y_j\|^2_{A(\lambda)^{-1}}}{(y_j^\top \theta_\star)^2} \log(1/2.4\delta).$$

Now let

$$\widetilde{\tau}^\star := \inf_{\lambda \in \Delta_\mathcal{X}} \max_{j=2,\dots,m} \frac{\|y_j\|^2_{A(\lambda)^{-1}}}{(y_j^\top \theta_\star)^2} \log(1/2.4\delta) = \max_{j=2,\dots,m} \frac{\|y_j\|^2_{A(\lambda^\star)^{-1}}}{(y_j^\top \theta_\star)^2} \log(1/2.4\delta),$$

where $\lambda^\star$ is the optimal design of the above optimization problem.[9] Set $\widetilde{t} = \widetilde{\tau}^\star \lambda^\star \in \mathbb{R}^n_+$ with $\widetilde{t}_i = \widetilde{\tau}^\star \lambda^\star_i \in \mathbb{R}_+$, we can then see that

$$\sum_{i=1}^{n} \widetilde{t}_i = \widetilde{\tau}^\star = \max_{j=2,\dots,m} \sum_{i=1}^{n} \widetilde{t}_i \frac{\|y_j\|^2_{A(\widetilde{t})^{-1}}}{(y_j^\top \theta_\star)^2} \log(1/2.4\delta), \forall\, 2 \leqslant j \leqslant m.$$

[9] Again, if the infimum is not attained, one can apply the argument that follows with a limit sequence. See footnote in Section 8.9.1.1 for more details on how to construct an approximating design.

and such $\{\widetilde{t}_i\}$ satisfies the constraints in the original optimization problem described in Eq. (8.15). As a result, we have $\tau^\star \leqslant \widetilde{\tau}^\star$.

We now can write

$$\tau^\star = \inf_{\lambda \in \Delta_{\mathcal{X}}} \max_{j=2,\dots,m} \frac{\|y_j\|^2_{A(\lambda)^{-1}}}{(y_j^\top \theta_\star)^2} \log(1/2.4\delta) = \rho^\star \log(1/2.4\delta). \tag{8.16}$$

Step 4: Monotonicity. We now apply the established equivalence to the model selection problem and prove monotonicity over $\{\rho^\star_d\}_{d=d_\star}^D$.

Now, define

$$\begin{aligned} \tau^\star_{d_\ell} = \min_{t_1,\dots,t_n \in \mathbb{R}_+} & \sum_{i=1}^n t_i \\ \text{s.t.} \inf_{\theta \in \mathcal{C}_{d_\ell}} & \sum_{i=1}^n t_i \, \mathrm{KL}(\nu_{\theta_\star,i}, \nu_{\theta,i}) \geqslant \log(1/2.4\delta), \end{aligned}$$

where $\mathcal{C}_{d_\ell} = \left\{\theta \in \mathbb{R}^D : \forall j > d_\ell : \theta_j = 0 \wedge \exists i \in [m] \text{ s.t. } \theta^\top (z_1 - z_i) < 0\right\}$. Let $d_\star \leqslant d_1 \leqslant d_2 \leqslant D$. Then, since the optimization problem in $\tau^\star_{d_1}$ has fewer constraints than the optimization problem in $\tau^\star_{d_2}$, we have that $\tau^\star_{d_1} \leqslant \tau^\star_{d_2}$. The established equivalence in Eq. (8.16) can be applied with respect to feature mappings $\psi_d(\cdot)$ for $d_\star \leqslant d \leqslant D$ (note that we necessarily have $\mathrm{span}(\{\psi_d(z_\star) - \psi_d(z)\}_{z \in \mathcal{Z} \setminus \{z_\star\}}) = \mathbb{R}^d$ as long as $\mathrm{span}(\{z_\star - z\}_{z \in \mathcal{Z} \setminus \{z_\star\}}) = \mathbb{R}^D$). Therefore, we have

$$\rho^\star_{d_1} \log(1/2.4\delta) = \tau^\star_{d_1} \leqslant \tau^\star_{d_2} = \rho^\star_{d_2} \log(1/2.4\delta),$$

leading to the desired result. □

8.9.2.4 Proof of Proposition 8.6

Proposition 8.6. *For any $\gamma > 0$, there exists an instance $(\mathcal{X}, \mathcal{Z}, \theta_\star \in \Theta_{d_\star})$ such that $\rho^\star_{d_\star+1} > \rho^\star_{d_\star} + \gamma$ yet $\iota^\star_{d_\star+1} \leqslant 2\iota^\star_{d_\star}$.*

Proof. For any $\lambda \in \Delta_{\mathcal{X}}$, we define

$$\rho_d(\lambda) := \max_{z \in \mathcal{Z} \setminus \{z_\star\}} \frac{\|\psi_d(z_\star) - \psi_d(z)\|^2_{A_d(\lambda)^{-1}}}{(h(z_\star) - h(z))^2},$$

and

$$\iota_d(\lambda) := \max_{z \in \mathcal{Z} \setminus \{z_\star\}} \|\psi_d(z_\star) - \psi_d(z)\|^2_{A_d(\lambda)^{-1}}.$$

We consider an instance $\mathcal{X} = \mathcal{Z} = \{x_i\}_{i=1}^{d_\star+1} \subseteq \mathbb{R}^{d_\star+1}$ and expected reward function $h(\cdot)$. The action set is constructed as follows:

$$x_i = e_i, \text{ for } i = 1, 2, \ldots, d_\star, \quad x_{d_\star+1} = (1-\varepsilon) \cdot e_{d_\star} + e_{d_\star+1},$$

where e_i is the i-th canonical basis in $\mathbb{R}^{d_\star+1}$. The expected reward of each action is set as

$$h(x_i) := \langle x_i, e_{d_\star} \rangle.$$

One can easily see that $d_\star$ is the intrinsic dimension of the problem (in fact, it is the smallest dimension such that linearity in rewards is preserved).

We notice that $\theta_\star \in \mathbb{R}^{d_\star}$; $x_\star = x_{d_\star}$ is the best arm with reward 1, $x_{d_\star+1}$ is the second best arm with reward $1-\varepsilon$ and all other arms have reward 0. The smallest sub-optimality gap is $\Delta_{\min} = \varepsilon$. $\varepsilon \in (0, 1/2]$ is selected such that $1/4\varepsilon^2 > 2d_\star + \gamma$ for any given $\gamma > 0$.[10]

We first consider truncating arms into $\mathbb{R}^{d_\star}$. For any $\lambda \in \Delta_{\mathcal{X}}$, we notice that $A_{d_\star}(\lambda) = \sum_{x \in \mathcal{X}} \lambda_x \psi_{d_\star}(x) \psi_{d_\star}(x)^\top$ is a diagonal matrix with the $d_\star$-th entry being $\lambda_{x_{d_\star}} + (1-\varepsilon)^2 \lambda_{x_{d_\star+1}}$ and the rest entries being λ_{x_i}. We first show that $\iota^\star_{d_\star} \geqslant d_\star - 1$ by contradiction as follows. Suppose $\iota^\star_{d_\star} < d_\star - 1$. Since $\|\psi_{d_\star}(x_\star) - \psi_{d_\star}(x_i)\|^2_{A_{d_\star}(\lambda)^{-1}} \geqslant 1/\lambda_{x_i}$ for $i = 1, 2, \ldots, d_\star - 1$, we must have $\lambda_{x_i} > 1/(d_\star - 1)$ for $i = 1, 2, \ldots, d_\star -$

[10] One can also add an additional arm $x_0 = e_D/2$ so that $\text{span}(\{x_\star - x\}_{x \in \mathcal{X}}) = \mathbb{R}^{d_\star+1}$ (the lower bound on $\rho^\star_{d_\star+1}$ will be changed to $1/16\varepsilon^2$).

1. Thus, $\sum_{i=1}^{d_\star-1}\lambda_{x_i} > 1$, which leads to a contradiction for $\lambda \in \Delta_{\mathcal{X}}$. We next analyze $\rho^\star_d$. Let $\lambda' \in \Delta_{\mathcal{X}}$ be the design such that $\lambda'_{x_i} = 1/d_\star$ for $i = 1, \ldots, d_\star$. With design λ', we have $\|\psi_{d_\star}(x_\star) - \psi_{d_\star}(x_i)\|^2_{A_{d_\star}(\lambda')^{-1}} = 2d_\star$ for $i = 1, 2, \ldots, d_\star - 1$ and $\|\psi_{d_\star}(x_\star) - \psi_{d_\star}(x_{d_\star+1})\|^2_{A_{d_\star}(\lambda')^{-1}} = \varepsilon^2 d_\star$. As a result, we have $\rho_{d_\star}(\lambda') \leqslant 2d_\star$, and thus $\rho^\star_{d_\star} \leqslant \rho_{d_\star}(\lambda') \leqslant 2d_\star$.

We now consider arms in the original space, i.e., $\mathbb{R}^{d_\star+1}$. We first upper bound $\iota^\star_{d_\star+1}$. With an uniform design λ'' such that $\lambda''_{x_i} = 1/(d_\star+1), \forall i \in [d_\star+1]$, we have $\iota^\star_{d_\star+1} \leqslant \iota_{d_\star+1}(\lambda'') \leqslant \max\{(3-\varepsilon)/(2-\varepsilon), \varepsilon^2/(2-\varepsilon)+1\} \cdot (d_\star+1) \leqslant 5(d_\star+1)/3$ when $\varepsilon \in (0, 1/2]$. In fact, with the same design, we can also upper bound $\iota(\mathcal{Y}(\psi_{d_\star+1}(\mathcal{X}))) \leqslant 3(d_\star+1)$. We analyze $\rho^\star_{d_\star+1}$ now. Since $\max_{x\in\mathcal{X}}\|x\|^2 \leqslant 4$ and $\min_{x\in\mathcal{X}\setminus\{x_\star\}}\|x_\star - x\|^2 \geqslant 1$, Lemma 8.19 leads to the fact that $\rho^\star_{d_\star+1} \geqslant 1/4\varepsilon^2$. Note that we only have $\min_{x\in\mathcal{X}\setminus\{x_\star\}}\|\psi_{d_\star}(x_\star) - \psi_{d_\star}(x)\|^2 \geqslant \varepsilon^2$ when truncating arms into $\mathbb{R}^{d_\star}$.

To summarize, for any given $\gamma > 0$, we have $\rho^\star_{d_\star+1} > \rho^\star_{d_\star} + \gamma$ yet $\iota^\star_{d_\star+1} \leqslant 2\iota^\star_{d_\star}$ (when $d_\star \geqslant 11$). Further more, we also have $\iota(\mathcal{Y}(\psi_{d_\star+1}(\mathcal{X}))) \leqslant 4\iota(\mathcal{Y}(\psi_{d_\star}(\mathcal{X})))$ (when $d_\star \geqslant 7$) since $\iota(\mathcal{Y}(\psi_{d_\star}(\mathcal{X}))) \leqslant \iota^\star_{d_\star}$. □

8.9.3 Proofs and Supporting Results for Section 8.4

8.9.3.1 Proof of Lemma 8.7

Lemma 8.7. *Suppose* $B \geqslant \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}$. *With probability at least* $1-\delta$, *GEMS-c outputs a set of arms* $\widehat{\mathcal{S}}_{n+1}$ *such that* $\Delta_z < 2^{1-n}$ *for any* $z \in \widehat{\mathcal{S}}_{n+1}$.

Proof. We consider event

$$\mathcal{E}_k = \{z_\star \in \widehat{\mathcal{S}}_k \subseteq \mathcal{S}_k\},$$

and prove through induction that

$$\mathbb{P}(\mathcal{E}_{k+1} \mid \cap_{i\leqslant k}\mathcal{E}_i) \geqslant 1 - \delta_k,$$

where $\delta_0 := 0$. Recall that $\mathcal{S}_k = \{z \in \mathcal{Z} : \Delta_z < 4 \cdot 2^{-k}\}$ (with $\mathcal{S}_1 = \mathcal{Z}$).

Step 1: The induction. We have $\left\{z_\star \in \widehat{\mathcal{S}}_1 \subseteq \mathcal{S}_1\right\}$ since $\widehat{\mathcal{S}}_1 = \mathcal{S}_1 = \mathcal{Z}$ by definition for the base case (recall that we assume $\max_{z\in\mathcal{Z}} \Delta_z \leqslant 2$). We now assume that $\cap_{i\leqslant k}\mathcal{E}_i$ holds true and we prove for iteration $k+1$. We only need to consider the case when $\left|\widehat{\mathcal{S}}_k\right| > 1$, which implies $|\mathcal{S}_k| > 1$ and thus $k \leqslant \left\lfloor \log_2(4/\Delta_{\min})\right\rfloor$.

Step 1.1: $d_k \geqslant d_\star$ **(Linearity is preserved).** Since $\widehat{\mathcal{S}}_k \subseteq \mathcal{S}_k$, we have

$$\begin{aligned} g_k(d_\star) &= \max\{2^{2k}\iota(\mathcal{Y}(\psi_{d_\star}(\widehat{\mathcal{S}}_k))), r_{d_\star}(\zeta)\} \\ &\leqslant \max\{2^{2k}\iota(\mathcal{Y}(\psi_{d_\star}(\mathcal{S}_k))), r_{d_\star}(\zeta)\} \\ &\leqslant \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\} \qquad (8.17) \\ &\leqslant B, \qquad (8.18) \end{aligned}$$

where Eq. (8.17) comes from Lemma 8.18 and Eq. (8.18) comes from the assumption. As a result, we know that $d_k \geqslant d_\star$ since d_k is selected as the largest integer such that $g_k(d_k) \leqslant B$.

Step 1.2: Concentration. Let $\{x_1, \ldots, x_{N_k}\}$ be the arms pulled at iteration k and $\{r_1, \ldots, r_{N_k}\}$ be the corresponding rewards. Let $\widehat{\theta}_k = A_k^{-1} b_k \in \mathbb{R}^{d_k}$ where $A_k = \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\psi_{d_k}(x_i)^\top$, and $b_k = \sum_{i=1}^{N_k} \psi_{d_k}(x_i) b_i$. Since $d_k \geqslant d_\star$ and the model is well-specified, we can write $r_i = \langle \theta_\star, x_i\rangle + \xi_i = \langle \psi_{d_k}(\theta_\star), \psi_{d_k}(x_i)\rangle + \xi_i$, where ξ_i is i.i.d. generated 1-sub-Gaussian noise. For any $y \in \mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k))$, we have

$$\begin{aligned} \left\langle y, \widehat{\theta}_k - \psi_{d_k}(\theta_\star)\right\rangle &= y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i) r_i - y^\top \psi_{d_k}(\theta_\star) \\ &= y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)(\psi_{d_k}(x_i)^\top \psi_{d_k}(\theta_\star) + \xi_i) - y^\top \psi_{d_k}(\theta_\star) \\ &= y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\xi_i. \end{aligned}$$

Since ξ_is are independent 1-sub-Gaussian random variables, we know that the random variable $y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\xi_i$ has variance proxy $\sqrt{\sum_{i=1}^{N_k}(y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i))^2} = \|y\|_{A_k^{-1}}$. Combining the standard Hoeffding's

inequality with a union bound leads to

$$\mathbb{P}\left(\forall y \in \mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k)), \left|\left\langle y, \widehat{\theta}_k - \psi_{d_k}(\theta_\star)\right\rangle\right| \leqslant \|y\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)}\right) \geqslant 1 - \delta_k, \tag{8.19}$$

where we use the fact that $|\mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k))| \leqslant |\widehat{\mathcal{S}}_k|^2/2$ in the union bound.

Step 1.3: Correctness. We prove $z_\star \in \widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$ under the good event analyzed in Eq. (8.19).

Step 1.3.1: $z_\star \in \widehat{\mathcal{S}}_{k+1}$**.** For any $\widehat{z} \in \widehat{\mathcal{S}}_k$ such that $\widehat{z} \neq z_\star$, we have

$$\begin{aligned}
&\langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \widehat{\theta}_k \rangle \\
&\leqslant \langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \psi_{d_k}(\theta_\star) \rangle + \|\psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star)\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)} \\
&= h(\widehat{z}) - h(z_\star) + \|\psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star)\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)} \\
&< \|\psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star)\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)}.
\end{aligned}$$

As a result, $z_\star$ remains in $\widehat{\mathcal{S}}_{k+1}$ according to the elimination criteria.

Step 1.3.2: $\widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$**.** Consider any $z \in \widehat{\mathcal{S}}_k \cap \mathcal{S}_{k+1}^c$, we know that $\Delta_z \geqslant 2 \cdot 2^{-k}$ by definition. Since $z_\star \in \widehat{\mathcal{S}}_k$, we then have

$$\begin{aligned}
&\langle \psi_{d_k}(z_\star) - \psi_{d_k}(z), \widehat{\theta}_k \rangle \\
&\geqslant \langle \psi_{d_k}(z_\star) - \psi_{d_k}(z), \psi_{d_k}(\theta_\star) \rangle - \|\psi_{d_k}(z_\star) - \psi_{d_k}(z)\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)} \\
&= h(z_\star) - h(z) - \|\psi_{d_k}(z_\star) - \psi_{d_k}(z)\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)} \\
&\geqslant 2 \cdot 2^{-k} - \|\psi_{d_k}(z_\star) - \psi_{d_k}(z)\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)} \\
&\geqslant \|\psi_{d_k}(z_\star) - \psi_{d_k}(z)\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)},
\end{aligned} \tag{8.20}$$

where Eq. (8.20) comes from the fact that $\|\psi_{d_k}(z_\star)-\psi_{d_k}(z)\|_{A_k^{-1}}\sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)} \leqslant 2^{-k}$, which is resulted from the choice of N_k and the guarantee in Eq. (8.3) from the rounding procedure. As a result, we have $z \notin \widehat{\mathcal{S}}_{k+1}$ and $\widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$.

To summarize, we prove the induction at iteration $k+1$, i.e.,

$$\mathbb{P}(\mathcal{E}_{k+1} \mid \cap_{i<k+1}\mathcal{E}_i) \geqslant 1-\delta_k.$$

Step 2: The error probability. Let $\mathcal{E} = \cap_{i=1}^{n+1}\mathcal{E}_i$ denote the good event, we then have

$$\begin{aligned}
\mathbb{P}(\mathcal{E}) &= \prod_{k=1}^{n}\mathbb{P}(\mathcal{E}_k \mid \mathcal{E}_{k-1}\cap\cdots\cap\mathcal{E}_1) \\
&= \prod_{k=1}^{n}(1-\delta_k) \\
&\geqslant \prod_{k=1}^{\infty}\left(1-\delta/k^2\right) \\
&= \frac{\sin(\pi\delta)}{\pi\delta} \\
&\geqslant 1-\delta, \qquad (8.21)
\end{aligned}$$

where we use the fact that $\sin(\pi\delta)/\pi\delta \geqslant 1-\delta$ for any $\delta \in (0,1)$ in Eq. (8.21). □

8.9.3.2 Proof of Theorem 8.8

Theorem 8.8. *Let* $\tau_\star = \log_2(4/\Delta_{\min})\max\left\{\rho^\star_{d_\star}, r_{d_\star}(\zeta)\right\}$. *With probability at least* $1-\delta$, *Algorithm 25 starts to output the optimal arm within iteration* $\ell_\star = O(\log_2(\tau_\star))$, *and takes at most* $N = O(\tau_\star\log_2(\tau_\star)\log(|\mathcal{Z}|\log_2(\tau_\star)/\delta))$ *samples.*

Proof. The proof is decomposed into three steps: (1) locating good subroutines; (2) bounding error probability and (3) bounding unverifiable sample complexity.

Step 1: Locating good subroutines. Consider $B_\star = \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}$ and $n_\star = \lceil\log_2(2/\Delta_{\min})\rceil$. For any subroutines invoked with $B_i \geqslant B_\star$ and $n_i \geqslant n_\star$, we

know that, from Lemma 8.7, the output set of arms are those with sub-optimality gap $< \Delta_{\min}$, which is a singleton set containing the optimal arm, i.e., $\{z_\star\}$. Let $i_\star = \lceil \log_2(B_\star) \rceil$, $j_\star = \lceil \log_2(n_\star) \rceil$ and $\ell_\star = i_\star + j_\star$. We know that in outer loops $\ell \geqslant \ell_\star$, there must exists at least one subroutine invoked with $B_i = 2^{i_\star} \geqslant B_\star$ and $n_i = 2^{j_\star} \geqslant n_\star$. Once a subroutine, invoked with $B_i \geqslant B_\star$, outputs a singleton set, it must be the optimal arm $z_\star$ according to Lemma 8.7 (up to small error probability, analyzed as below). Since, within each outer loop ℓ, the value of $B_i = 2^{\ell - i}$ is chosen in a decreasing order, updating the recommendation and breaking the inner loop once a singleton set is identified will not miss the chance of recommending the optimal arm in later subroutines within outer loop ℓ.

Step 2: Error probability. We consider the good event where all subroutines invoked in Algorithm 25 with $B_i \geqslant B_\star$ and (any) n_i correctly output a set of arms with sub-optimality gap $< 2^{1-n_i}$ with probability at least $1 - \delta_\ell$, as shown in Lemma 8.7. This good event clearly happens with probability at least $1 - \sum_{\ell=1}^{\infty} \sum_{i=1}^{\ell} \delta_\ell = 1 - \sum_{\ell=1}^{\infty} \delta/(2\ell^2) > 1 - \delta$, after applying a union bound argument. We upper bound the unverifiable sample complexity under this event in the following.

Step 3: Unverifiable sample complexity. For any subroutine invoked within outer loop $\ell \leqslant \ell_\star$, we know, from Algorithm 26, that its sample complexity is upper bounded by (note that $|\mathcal{Z}|^2 \geqslant 4$ trivially holds true)

$$\begin{aligned} N_\ell &\leqslant n_i \big(B_i \cdot \big(2.5 \log(|\mathcal{Z}|^2/\delta_{\ell_\star}) \big) + 1 \big) \\ &\leqslant \gamma_\ell \, 3.5 \log\big(2|\mathcal{Z}|^2 \ell_\star^3/\delta\big). \end{aligned}$$

Thus, the total sample complexity up to the end of outer loop $\ell_\star$ is upper bounded by

$$\begin{aligned} N &\leqslant \sum_{\ell=1}^{\ell_\star} \ell N_\ell \\ &\leqslant 3.5 \log\big(2|\mathcal{Z}|^2 \ell_\star^3/\delta\big) \sum_{\ell=1}^{\ell_\star} \ell 2^\ell \end{aligned}$$

$$\leqslant 7 \log\bigl(2|\mathcal{Z}|^2 \ell_\star^3/\delta\bigr) \ell_\star 2^{\ell_\star}.$$

Recall that $\tau_\star = \log_2(4/\Delta_{\min}) \max\bigl\{\rho^\star_{d_\star}, r_{d_\star}(\zeta)\bigr\}$. By definition of $\ell_\star$, we have

$$\ell_\star \leqslant \log_2\bigl(4 \log_2(4/\Delta_{\min}) \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}\bigr) = O(\log_2(\tau_\star)),$$

and

$$\begin{aligned} 2^{\ell_\star} &= 2^{(i_\star + j_\star)} \\ &\leqslant 4\bigl(\log_2(2/\Delta_{\min}) + 1\bigr) \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}, \\ &= 4 \log_2(4/\Delta_{\min}) \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}, \\ &= O(\tau_\star). \end{aligned}$$

The unverifiable sample complexity is thus upper bounded by

$$\begin{aligned} N &\leqslant 1792\, \tau_\star \cdot \bigl(\log_2(\tau_\star) + 8\bigr) \cdot \log\bigl(2|\mathcal{Z}|^2 (\log_2(\tau_\star) + 8)^3/\delta\bigr) \\ &= O\bigl(\tau_\star \log_2(\tau_\star) \log(|\mathcal{Z}| \log_2(\tau_\star)/\delta)\bigr). \end{aligned}$$

□

8.9.4 Proofs and Supporting Results for Section 8.5

8.9.4.1 Proof of Lemma 8.9

Lemma 8.9. *Suppose* $64\rho^\star_{d_\star} \leqslant B \leqslant 128\rho^\star_{d_\star}$ *and* $T/n \geqslant r_{d_\star}(\zeta) + 1$. *Algorithm 26 outputs an arm* $\widehat{z}_\star$ *such that* $\Delta_{\widehat{z}_\star} < 2^{1-n}$ *with probability at least*

$$1 - n|\mathcal{Z}|^2 \exp\bigl(-T/640\, n\, \rho^\star_{d_\star}\bigr).$$

Proof. We consider event

$$\mathcal{E}_k = \{z_\star \in \widehat{S}_k \subseteq S_k\},$$

and prove through induction that

$$\mathbb{P}(\mathcal{E}_{k+1} \mid \cap_{i \leqslant k} \mathcal{E}_i) \geqslant 1 - \delta_k,$$

where the value of $\{\delta_k\}_{k=0}^n$ will be specified in the proof.

Step 1: The induction. The base case $\{z_\star \in \widehat{\mathcal{S}}_1 \subseteq \mathcal{S}_1\}$ holds with probability 1 by construction (thus, we have $\delta_0 = 0$). Conditioned on events $\cap_{i=1}^k \mathcal{E}_i$, we next analyze the event $\mathcal{E}_{k+1}$. We only need to consider the case when $|\widehat{\mathcal{S}}_k| > 1$, which implies $|\mathcal{S}_k| > 1$ and thus $k \leqslant \lfloor \log_2(4/\Delta_{\min}) \rfloor$.

Step 1.1: $d_k \geqslant d_\star$ (Linearity is preserved). We first notice that $\widetilde{D}$ is selected as the largest integer such that $r_{\widetilde{D}}(\zeta) \leqslant T'$, where $r_d(\zeta)$ represents the number of samples needed for the rounding procedure in $\mathbb{R}^d$ (with parameter ζ). When $T/n \geqslant r_{d_\star}(\zeta) + 1$, we have $\widetilde{D} \geqslant d_\star$ since $T' \geqslant T/n - 1 \geqslant r_{d_\star}(\zeta)$. Thus, for whatever $d_k \in [\widetilde{D}]$ selected, we always have $r_{d_k}(\zeta) \leqslant r_{\widetilde{D}}(\zeta) \leqslant T'$ and can thus safely apply the rounding procedure described in Eq. (8.3).

Since $\widehat{\mathcal{S}}_k \subseteq \mathcal{S}_k$, we also have

$$\begin{aligned} g_k(d_\star) &= 2^{2k} \iota(\mathcal{Y}(\psi_{d_\star}(\widehat{\mathcal{S}}_k))) \\ &\leqslant 2^{2k} \iota(\mathcal{Y}(\psi_{d_\star}(\mathcal{S}_k))) \\ &\leqslant 64 \rho^\star_{d_\star} \qquad (8.22) \\ &\leqslant B, \qquad (8.23) \end{aligned}$$

where Eq. (8.22) comes from Lemma 8.18 and Eq. (8.23) comes from the assumption. As a result, we know that $d_k \geqslant d_\star$ since $d_k \in [\widetilde{D}]$ is selected as the largest integer such that $g_k(d_k) \leqslant B$.

Step 1.2: Concentration and error probability. Let $\{x_1, \ldots, x_{T'}\}$ be the arms pulled at iteration k and $\{r_1, \ldots, r_{T'}\}$ be the corresponding rewards. Let $\widehat{\theta}_k = A_k^{-1} b_k \in \mathbb{R}^{d_k}$ where $A_k = \sum_{i=1}^{T'} \psi_{d_k}(x_i) \psi_{d_k}(x_i)^\top$, and $b_k = \sum_{i=1}^{T'} \psi_{d_k}(x_i) b_i$. Since $d_k \geqslant d_\star$ and the model is well-specified, we can write $r_i = \langle \theta_\star, x_i \rangle + \xi_i = \langle \psi_{d_k}(\theta_\star), \psi_{d_k}(x_i) \rangle + \xi_i$, where ξ_i is i.i.d. generated zero-mean Gaussian noise with

variance 1. Similarly as analyzed in Eq. (8.19), we have

$$\mathbb{P}\left(\forall y \in \mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k)), \left|\left\langle y, \widehat{\theta}_k - \psi_{d_k}(\theta_\star)\right\rangle\right| \leqslant \|y\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)}\right) \geqslant 1-\delta_k. \tag{8.24}$$

By setting $\max_{y\in\psi_{d_k}(\widehat{\mathcal{S}}_k)}\|y\|_{A_k^{-1}}\sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)} = 2^{-k}$, we have

$$\begin{aligned}
\delta_k &= |\widehat{\mathcal{S}}_k|^2 \exp\left(-\frac{1}{2\cdot 2^{2k}\ \max_{y\in\psi_{d_k}(\widehat{\mathcal{S}}_k)}\|y\|^2_{A_k^{-1}}}\right) \\
&\leqslant |\widehat{\mathcal{S}}_k|^2 \exp\left(-\frac{T'}{2\cdot 2^{2k}\,(1+\zeta)\,\iota(\mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k)))}\right) &(8.25)\\
&\leqslant |\mathcal{Z}|^2 \exp\left(-\frac{T}{1024\,n\,\rho^\star_{d_\star}}\right), &(8.26)
\end{aligned}$$

where Eq. (8.25) comes from the guarantee of the rounding procedure Eq. (8.3); and Eq. (8.26) comes from combining the following facts: (1) $2^{2k}\,\iota(\mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k))) \leqslant B \leqslant 128\rho^\star_{d_\star}$; (2) $T' \geqslant T/n - 1 \geqslant T/2n$ (note that $T/n \geqslant r_{d_\star}(\zeta)+1 \implies T/n \geqslant 2$ since $r_{d_\star}(\zeta) \geqslant 1$); (3) $\widehat{\mathcal{S}}_k \subseteq \mathcal{Z}$ and (4) consider some $\zeta \leqslant 1$ (ζ only affects constant terms).

Step 1.3: Correctness. We prove $z_\star \in \widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$ under the good event analyzed in Eq. (8.24).

Step 1.3.1: $z_\star \in \widehat{\mathcal{S}}_{k+1}$**.** For any $\widehat{z} \in \widehat{\mathcal{S}}_k$ such that $\widehat{z} \neq z_\star$, we have

$$\begin{aligned}
\langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \widehat{\theta}_k\rangle &\leqslant \langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \psi_{d_k}(\theta_\star)\rangle + 2^{-k} \\
&= h(\widehat{z}) - h(z_\star) + 2^{-k} \\
&< 2^{-k}.
\end{aligned}$$

As a result, $z_\star$ remains in $\widehat{\mathcal{S}}_{k+1}$ according to the elimination criteria.

Step 1.3.2: $\widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$**.** Consider any $z \in \widehat{\mathcal{S}}_k \cap \mathcal{S}^c_{k+1}$, we know that $\Delta_z \geqslant 2\cdot 2^{-k}$

by definition. Since $z_\star \in \widehat{\mathcal{S}}_k$, we then have

$$\begin{aligned}\langle \psi_{d_k}(z_\star) - \psi_{d_k}(z), \widehat{\theta}_k \rangle &\geqslant \langle \psi_{d_k}(z_\star) - \psi_{d_k}(z), \psi_{d_k}(\theta_\star) \rangle - 2^{-k} \\ &= h(z_\star) - h(z) - 2^{-k} \\ &\geqslant 2 \cdot 2^{-k} - 2^{-k} \\ &= 2^{-k}. \end{aligned} \tag{8.27}$$

As a result, we have $z \notin \widehat{\mathcal{S}}_{k+1}$ and $\widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$.

To summarize, we prove the induction at iteration $k+1$, i.e.,

$$\mathbb{P}(\mathcal{E}_{k+1} \mid \cap_{i<k+1} \mathcal{E}_i) \geqslant 1 - \delta_k.$$

Step 2: The error probability. Let $\mathcal{E} = \cap_{i=1}^{n+1} \mathcal{E}_i$ denote the good event, we then have

$$\begin{aligned}\mathbb{P}(\mathcal{E}) &= \prod_{k=1}^{n+1} \mathbb{P}(\mathcal{E}_k \mid \mathcal{E}_{k-1} \cap \cdots \cap \mathcal{E}_1) \\ &= \prod_{k=1}^{n+1} (1 - \delta_k) \\ &\geqslant 1 - \sum_{i=1}^{n+1} \delta_k \qquad (8.28) \\ &\geqslant 1 - n|\mathcal{Z}|^2 \exp\left(-\frac{T}{640\, n\, \rho^\star_{d_\star}}\right), \end{aligned}$$

where Eq. (8.28) can be proved using a simple induction. □

8.9.4.2 Proof of Theorem 8.10

Theorem 8.10. *Suppose $\mathcal{Z} \subseteq \mathcal{X}$. If $T = \widetilde{\Omega}(\log_2(1/\Delta_{\min}) \max\{\rho^\star_{d_\star}, r_{d_\star}(\zeta)\})$, then Algorithm 27 outputs the optimal arm with error probability at most*

$$\log_2(4/\Delta_{\min})|\mathcal{Z}|^2 \exp\left(-\frac{T}{1024 \log_2(4/\Delta_{\min})\, \rho^\star_{d_\star}}\right) + 2(\log_2 T)^2 \exp\left(-\frac{T}{8(\log_2 T)^2/\Delta^2_{\min}}\right).$$

Furthermore, if there exist universal constants such that $\max_{x\in\mathcal{X}}\|\psi_{d_\star}(x)\|^2 \leqslant c_1$ and $\min_{z\in\mathcal{Z}}\|\psi_{d_\star}(z_\star) - \psi_{d_\star}(z)\|^2 \geqslant c_2$, the error probability is upper bounded by

$$O\left(\max\left\{\log_2(1/\Delta_{\min})|\mathcal{Z}|^2, (\log_2 T)^2\right\} \times \exp\left(-\frac{c_2 T}{\max\{\log_2(1/\Delta_{\min}), (\log_2 T)^2\} c_1 \rho^\star_{d_\star}}\right)\right).$$

Proof. The proof is decomposed into three steps: (1) locate a good subroutine in the pre-selection step; (2) bound error probability in the validation step; and (3) analyze the total error probability. Some preliminaries are analyzed as follows.

We note that both pre-selection and validation steps use budget less than T: in the pre-selection phase, each outer loop indexed by i uses budget less than T/p and there are p such outer loops; it's also clear that the validation steps uses at most T budget. We notice that $p \leqslant \log_2 T$ since $p \cdot 2^p \leqslant T$; and $q_i \leqslant \log_2 T$ since $q_i \cdot 2^{q_i} \leqslant T/pB_i \leqslant T$. As a result, at most $(\log_2 T)^2$ subroutines are invoked in Algorithm 27, and each subroutine is invoked with budget $T'' \geqslant T/(\log_2 T)^2$.

Step 1: The good subroutines. Consider

$$i_\star := \lceil \log_2(64\rho^\star_{d_\star}) \rceil \quad \text{and} \quad j_\star := \lceil \log_2(\log_2(2/\Delta_{\min})) \rceil.$$

One can easily see that $64\rho^\star_{d_\star} \leqslant B_{i_\star} \leqslant 128\rho^\star_{d_\star}$ and $n_{j_\star} \geqslant \log_2(2/\Delta_{\min})$. Thus, once a subroutine is invoked with $(i_\star, j_\star)$ and $T''/n_{j_\star} \geqslant r_{d_\star}(\zeta) + 1$, Lemma 8.9 guarantees

to output the optimal arm with error probability at most

$$\log_2(4/\Delta_{\min})|\mathcal{Z}|^2 \exp\left(-\frac{T}{1024\,\log_2(4/\Delta_{\min})\,\rho^\star_{d_\star}}\right). \tag{8.29}$$

We next show that for sufficiently large T, one can invoke the subroutine with $(i_\star, j_\star)$ and $T''/n_{j_\star} \geqslant r_{d_\star}(\zeta) + 1$.

We clearly have $p \geqslant i_\star$ as long as $T \geqslant \log_2(128\rho^\star_{d_\star})\,128\rho^\star_{d_\star}$. Focusing on the outer loop with index $i_\star$, we have $q_{i_\star} \geqslant j_\star$ as long as

$$\log_2(2\log_2(2/\Delta_{\min})) \cdot (2\log_2(2/\Delta_{\min})) \leqslant T'/B_{i_\star},$$

Since $T'/B_{i_\star} \geqslant T/(128\rho^\star_{d_\star}\log_2 T)$, we have $q_{i_\star} \geqslant j_\star$ as long as T is such that

$$T \geqslant 256\,\log_2(2\log_2(2/\Delta_{\min})) \cdot \log_2(2/\Delta_{\min}) \cdot \rho^\star_{d_\star} \cdot \log_2 T. \tag{8.30}$$

Since $T'' \geqslant T/(\log_2 T)^2$, we have $T''/n_{j_\star} \geqslant r_{d_\star}(\zeta) + 1$ as long as T is such that

$$T \geqslant (r_{d_\star}(\zeta) + 1) \cdot \log_2(4/\Delta_{\min}) \cdot (\log_2 T)^2. \tag{8.31}$$

According to Lemma 8.20, Eq. (8.30) and Eq. (8.31) can be satisfied when

$$T = \widetilde{\Omega}\big(\log_2(1/\Delta_{\min})\max\{\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}\big),$$

where lower order terms with respect to $\log_2(1/\Delta_{\min})$, $\rho^\star_{d_\star}$ and $r_{d_\star}(\zeta)$ are hidden in the $\widetilde{\Omega}$ notation.

Step 2: The validation step. We have $|\mathcal{A}| \leqslant (\log_2 T)^2$ since there are at most $(\log_2 T)^2$ subroutines and each subroutine outputs one arm. We view each $x \in \mathcal{A}$ as individual arm and pull it $\lfloor T/|\mathcal{A}|\rfloor \geqslant T/(\log_2 T)^2 - 1 \geqslant T/2(\log_2 T)^2$ (as long as $T \geqslant 2(\log_2 T)^2$) times. We use $\widehat{h}(x)$ to denote the empirical mean of $h(x)$. Applying Hoeffding's inequality with a union bound leads to the following concentration

result

$$\mathbb{P}\Big(\forall x \in \mathcal{A} : |\widehat{h}(x) - h(x)| \geqslant \Delta_{\min}/2\Big) \leqslant 2(\log_2 T)^2 \exp\left(-\frac{T}{8(\log_2 T)^2/\Delta_{\min}^2}\right)$$

Thus, as long as $z_\star \in \mathcal{A}$ is selected in $\mathcal{A}$ from the pre-selection step, the validation step correctly output $z_\star$ with error probability at most

$$2(\log_2 T)^2 \exp\left(-\frac{T}{8(\log_2 T)^2/\Delta_{\min}^2}\right). \tag{8.32}$$

Step 3: Total error probability. Combining Eq. (8.29) with Eq. (8.32), we know that

$$\begin{aligned}\mathbb{P}(\widehat{z}_\star \neq z_\star) \leqslant & \log_2(4/\Delta_{\min})|\mathcal{Z}|^2 \exp\left(-\frac{T}{1024\ \log_2(4/\Delta_{\min})\ \rho^\star_{d_\star}}\right) \\ & + 2(\log_2 T)^2 \exp\left(-\frac{T}{8(\log_2 T)^2/\Delta_{\min}^2}\right).\end{aligned}$$

Furthermore, if there exists universial constants such that $\max_{x\in\mathcal{X}}\|\psi_{d_\star}(x)\|^2 \leqslant c_1$ and $\min_{z\in\mathcal{Z}}\|\psi_{d_\star}(z_\star) - \psi_{d_\star}(z)\|^2 \geqslant c_2$, Lemma 8.19 implies that $1/\Delta_{\min}^2 \leqslant c_1\rho^\star_{d_\star}/c_2$. We thus have

$$\mathbb{P}(\widehat{z}_\star \neq z_\star) = O\left(\max\Big\{\log_2(1/\Delta_{\min})|\mathcal{Z}|^2, (\log_2 T)^2\Big\} \cdot \exp\left(-\frac{c_2 T}{\max\big\{\log_2(1/\Delta_{\min}), (\log_2 T)^2\big\} c_1 \rho^\star_{d_\star}}\right)\right).$$

□

8.9.5 Proofs and Supporting Results for Section 8.6

8.9.5.1 Proofs for Propositions

Some of the propositions are borrowed from Zhu et al. (2021), we present detailed proofs here for completeness.

Proposition 8.11. *The misspecification level $\widetilde{\gamma}(d)$ is non-increasing with respect to* d.

Proof. Consider any $1 \leqslant d < d' \leqslant D$. Suppose

$$\theta^d \in \arg\min_{\theta \in \mathbb{R}^D} \max_{x \in \mathcal{X} \cup \mathcal{Z}} |h(x) - \langle \psi_d(\theta), \psi_d(x) \rangle|.$$

Since $\psi_d(\theta^d)$ only keeps the first d component of θ^d, we can choose θ^d such that it only has non-zero values on its first d entries. As a result, we have $\langle \psi_d(\theta^d), \psi_d(x) \rangle = \langle \psi_{d'}(\theta^d), \psi_{d'}(x) \rangle$, which implies that $\widetilde{\gamma}(d') \leqslant \widetilde{\gamma}(d)$. □

Proposition 8.12 (Zhu et al. (2021))**.** *We have $\rho^\star_d(\varepsilon) \leqslant 9\widetilde{\rho}^\star_d(\varepsilon)$ for any $\varepsilon \geqslant \widetilde{\gamma}(d)$. Furthermore, if $\widetilde{\gamma}(d) < \Delta_{\min}/2$, $\widetilde{\rho}^\star_d(0)$ represents the complexity measure for best arm identification with respect to a linear bandit instance with action set $\mathcal{X}$, target set $\mathcal{Z}$ and reward function $\widetilde{h}(x) := \langle \psi_d(\theta^d_\star), \psi_d(x) \rangle$.*

Proof. To relate $\rho^\star_d(\varepsilon)$ with $\widetilde{\rho}^\star_d(\varepsilon)$, we only need to relate $\max\{h(z_\star) - h(z), \varepsilon\}$ with $\max\{\langle \psi_d(z_\star) - \psi_d(z), \theta^d_\star \rangle, \varepsilon\}$. From Eq. (8.5) and the fact that $\varepsilon \geqslant \widetilde{\gamma}(d)$, we know that

$$\begin{aligned}
\langle \psi_d(z_\star) - \psi_d(z), \theta^d_\star \rangle &\leqslant h(z_\star) - h(z) + 2\widetilde{\gamma}(d) \\
&\leqslant h(z_\star) - h(z) + 2\varepsilon \\
&\leqslant 3\max\{h(z_\star) - h(z), \varepsilon\},
\end{aligned}$$

and thus

$$\max\left\{\langle \psi_d(z_\star) - \psi_d(z), \theta^d_\star \rangle, \varepsilon\right\} \leqslant 3\max\{h(z_\star) - h(z), \varepsilon\}.$$

As a result, we have $\rho^\star_d(\varepsilon) \leqslant 9\widetilde{\rho}^\star_d(\varepsilon)$.

When $\widetilde{\gamma}(d) < \Delta_{\min}/2$, we know that $z_\star$ is still the best arm in the perfect linear bandit model (without misspecification) $\widetilde{h}(x) = \langle \psi_d(x), \psi_d(\theta^d_\star) \rangle$. Thus, $\widetilde{\rho}^\star_d(0)$ represents the complexity measure, in the corresponding linear model, for best arm identification. □

Proposition 8.21 (Zhu et al. (2021)). *The following inequalities hold:*

$$\gamma(d) \leqslant \left(16 + 16\sqrt{(1+\zeta)d}\right)\widetilde{\gamma}(d) = O(\sqrt{d}\,\widetilde{\gamma}(d)).$$

Proof. We first notice that

$$\begin{aligned}
\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k))) &= \inf_{\lambda\in\Delta_{\mathcal{X}}} \sup_{y\in\mathcal{Y}(\psi_d(\mathcal{S}_k))} \|y\|^2_{A_d(\lambda)^{-1}} \\
&\leqslant \inf_{\lambda\in\Delta_{\mathcal{X}}} \sup_{y\in\mathcal{Y}(\psi_d(\mathcal{X}))} \|y\|^2_{A_d(\lambda)^{-1}} \\
&\leqslant \inf_{\lambda\in\Delta_{\mathcal{X}}} \sup_{x\in\mathcal{X}} 4\|\psi_d(x)\|^2_{A_d(\lambda)^{-1}} \\
&= 4d, \qquad (8.33)
\end{aligned}$$

where Eq. (8.33) comes from Kiefer-Wolfowitz theorem (Kiefer and Wolfowitz, 1960). We then have

$$(2+\sqrt{(1+\zeta)\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k)))})\widetilde{\gamma}(d) \leqslant (2+\sqrt{(1+\zeta)4d})\widetilde{\gamma}(d).$$

As a result, we can always find a $n \in \mathbb{N}$ such that

$$2^{-n}/2 \leqslant 2\,(2+\sqrt{(1+\zeta)4d})\widetilde{\gamma}(d),$$

and

$$(2+\sqrt{(1+\zeta)\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k)))})\widetilde{\gamma}(d) \leqslant (2+\sqrt{(1+\zeta)4d})\widetilde{\gamma}(d) \leqslant 2^{-k}/2, \forall k \leqslant n.$$

This leads to the fact that

$$\gamma(d) \leqslant 8\,(2+\sqrt{(1+\zeta)4d})\widetilde{\gamma}(d),$$

which implies the desired result. □

Proposition 8.22. *If $\gamma(d) \leqslant \varepsilon$, we have*

$$(2+\sqrt{(1+\zeta)\iota(\mathcal{Y}(\psi_d(\mathcal{S}_k)))})\widetilde{\gamma}(d) \leqslant 2^{-k}/2, \forall k \leqslant \left\lceil \log_2(2/\varepsilon) \right\rceil.$$

Proof. Suppose $\gamma(d) = 2 \cdot 2^{-\widetilde{n}}$ for a $\widetilde{n} \in \mathbb{N}$. Since $\gamma(d) \leqslant \varepsilon$, we have $\widetilde{n} \geqslant \log_2(2/\varepsilon)$. Since $\widetilde{n} \in \mathbb{N}$, we know that $\widetilde{n} \geqslant \left\lceil \log_2(2/\varepsilon) \right\rceil$. The desired result follows from the definition of $\gamma(d)$. □

8.9.5.2 Omitted Details for the Fixed Confidence Setting with Misspecification

Omitted Algorithms.

Algorithm 29 GEMS-m Gap Elimination with Model Selection with Misspecification (Fixed Confidence)

Input: Number of iterations n, budget for dimension selection B and confidence parameter δ.

1: Set $\widehat{\mathcal{S}}_1 = \mathcal{Z}$.
2: **for** $k = 1, 2, \ldots, n$ **do**
3: Set $\delta_k = \delta/k^2$.
4: Define function $g_k(d) := \max\{2^{2k}\iota_{k,d}, r_d(\zeta)\}$, where $\iota_{k,d} := \iota(\mathcal{Y}(\psi_d(\widehat{\mathcal{S}}_k)))$.
5: Get $d_k = \mathsf{OPT}(B, D, g_k(\cdot))$, where $d_k \leqslant D$ is largest dimension such that $g_k(d_k) \leqslant B$ (see Eq. (8.4) for the detailed optimization problem). Set λ_k be the optimal design of the optimization problem $\inf_{\lambda \in \Delta_{\mathcal{X}}} \sup_{z,z' \in \widehat{\mathcal{S}}_k} \|\psi_{d_k}(z) - \psi_{d_k}(z')\|^2_{A_{d_k}(\lambda)^{-1}}$; set $N_k = \lceil g(d_k) 8(1+\zeta) \log(|\widehat{\mathcal{S}}_k|^2/\delta_k) \rceil$.
6: Get allocation $\{x_1, \ldots, x_{N_k}\} = \mathsf{ROUND}(\lambda_k, N_k, d_k, \zeta)$.
7: Pull arms $\{x_1, \ldots, x_{N_k}\}$ and receive rewards $\{r_1, \ldots, r_{N_k}\}$.
8: Set $\widehat{\theta}_k = A_k^{-1} b_k \in \mathbb{R}^{d_k}$ where $A_k = \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\psi_{d_k}(x_i)^\top$, and $b_k = \sum_{i=1}^{N_k} \psi_{d_k}(x_i) b_i$.
9: Set $\widehat{\mathcal{S}}_{k+1} = \widehat{\mathcal{S}}_k \setminus \{z \in \widehat{\mathcal{S}}_k : \exists z' \text{ s.t. } \langle \widehat{\theta}_k, \psi_{d_k}(z') - \psi_{d_k}(z) \rangle \geqslant 2^{-k}\}$.

Output: Any $\widehat{z}_\star \in \widehat{\mathcal{S}}_{n+1}$ (or the whole set $\widehat{\mathcal{S}}_{n+1}$ when aiming at identifying the optimal arm).

Algorithm 30 Adaptive Strategy for Model Selection with misspecification (Fixed Confidence)

Input: Confidence parameter δ.

1: Randomly select a $\widehat{z}_\star \in \mathcal{X}$ as the recommendation for the ε-optimal arm.
2: **for** $\ell = 1, 2, \ldots$ **do**
3: Set $\gamma_\ell = 2^\ell$ and $\delta_\ell = \delta/(4\ell^3)$. Initialize an empty pre-selection set $\mathcal{A}_\ell = \{\}$.
4: **for** $i = 1, 2, \ldots, \ell$ **do**
5: Set $n_i = 2^i$, $B_i = 2^{\ell - i}$ and get $\widehat{z}_\star^i = \mathsf{GEMS\text{-}m}(n_i, B_i, \delta_\ell)$. Insert $\widehat{z}_\star^i$ into $\mathcal{A}_\ell$.
6: **Validation.** Pull each arm in $\mathcal{A}$ exactly $\lceil 8 \log(2/\delta_\ell)/\varepsilon^2 \rceil$ times. Update $\widehat{z}_\star$ as the arm with the highest empirical mean (break ties arbitrarily).

Lemma 8.23 and Its Proof.

We introduce function $f : \mathbb{N}_+ \to \mathbb{R}_+$ as follows, which is also used in Section 8.9.5.4.

$$f(k) := \begin{cases} 4 \cdot 2^{-k} & \text{if } k \leqslant \lceil \log_2(2/\varepsilon) \rceil + 1, \\ 4 \cdot \varepsilon^{-\lceil \log_2(4/\varepsilon) \rceil} & \text{if } k > \lceil \log_2(2/\varepsilon) \rceil + 1. \end{cases}$$

$f(k)$ is used to quantify the optimality of the identified arm, and one can clearly see that $f(k)$ is non-increasing in k.

Lemma 8.23. *Suppose* $B \geqslant \max\{64\rho^\star_{d_\star(\varepsilon)}(\varepsilon), r_{d_\star(\varepsilon)}(\zeta)\}$. *With probability at least* $1 - \delta$, *Algorithm 29 outputs an arm* $\widehat{z}_\star$ *such that* $\Delta_{\widehat{z}_\star} < f(n+1)$. *Furthermore, an* ε*-optimal arm is output as long as* $n \geqslant \log_2(2/\varepsilon)$.

Proof. The logic of this proof is similar to the proof of Lemma 8.7. We additionally deal with misspecification in the proof. For fixed ε, we use the notation $d_\star = d_\star(\varepsilon)$ throughout the proof.

We consider event

$$\mathcal{E}_k = \{z_\star \in \widehat{\mathcal{S}}_k \subseteq \mathcal{S}_k\},$$

and prove through induction that, for $k \leqslant \lceil \log_2(2/\varepsilon) \rceil$,

$$\mathbb{P}(\mathcal{E}_{k+1} \mid \cap_{i \leqslant k} \mathcal{E}_i) \geqslant 1 - \delta_k,$$

where $\delta_0 := 0$. Recall that $\mathcal{S}_k = \{z \in \mathcal{Z} : \Delta_z < 4 \cdot 2^{-k}\}$ (with $\mathcal{S}_1 = \mathcal{Z}$). For $n \geqslant k+1$, we have $\widehat{\mathcal{S}}_n \subseteq \widehat{\mathcal{S}}_{k+1}$ due to the nature of the elimination-styled algorithm, which guarantees outputting an arm such that $\Delta_z < f(n+1)$.

Step 1: The induction. We have $\{z_\star \in \widehat{\mathcal{S}}_1 \subseteq \mathcal{S}_1\}$ since $\widehat{\mathcal{S}}_1 = \mathcal{S}_1 = \mathcal{Z}$ by definition for the base case (recall we assume that $\max_{z \in \mathcal{Z}} \Delta_z \leqslant 2$). We now assume that $\cap_{i<k+1} \mathcal{E}_i$ holds true and we prove for iteration $k+1$.

Step 1.1: $d_k \geqslant d_\star$**.** Since $\widehat{\mathcal{S}}_k \subseteq \mathcal{S}_k$, we have

$$\begin{aligned}
g_k(d_\star) &= \max\{2^{2k} \iota(\mathcal{Y}(\psi_{d_\star}(\widehat{\mathcal{S}}_k))), r_{d_\star}(\zeta)\} \\
&\leqslant \max\{2^{2k} \iota(\mathcal{Y}(\psi_{d_\star}(\mathcal{S}_k))), r_{d_\star}(\zeta)\} \\
&\leqslant \max\{64 \rho^\star_{d_\star}(\varepsilon), r_{d_\star}(\zeta)\} \qquad (8.34) \\
&\leqslant B, \qquad (8.35)
\end{aligned}$$

where Eq. (8.34) comes from Lemma 8.17 and Eq. (8.35) comes from the assumption. As a result, we know that $d_k \geqslant d_\star$ since d_k is selected as the largest integer such that $g_k(d_k) \leqslant B$.

Step 1.2: Concentration. Let $\{x_1, \ldots, x_{N_k}\}$ be the arms pulled at iteration k and $\{r_1, \ldots, r_{N_k}\}$ be the corresponding rewards. Let $\widehat{\theta}_k = A_k^{-1} b_k \in \mathbb{R}^{d_k}$ where $A_k = \sum_{i=1}^{N_k} \psi_{d_k}(x_i) \psi_{d_k}(x_i)^\top$, and $b_k = \sum_{i=1}^{N_k} \psi_{d_k}(x_i) b_i$. Based on the definition of $\theta^d_\star \in \mathbb{R}^D$ and $\eta_d(\cdot)$, we can write $r_i = h(x_i) + \xi_i = \left\langle \psi_{d_k}(\theta^{d_k}_\star), \psi_{d_k}(x_i) \right\rangle + \eta_{d_k}(x_i) + \xi_i$, where ξ_i is i.i.d. generated zero-mean Gaussian noise with variance 1; we also have $|\eta_{d_k}(x_i)| \leqslant \widetilde{\gamma}(d_k)$ by definition of $\widetilde{\gamma}(\cdot)$. For any $y \in \mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k))$, we have

$$\begin{aligned}
&\left| \langle y, \widehat{\theta}_k - \psi_{d_k}(\theta^{d_k}_\star) \rangle \right| \\
&= \left| y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i) r_i - y^\top \psi_{d_k}(\theta^{d_k}_\star) \right|
\end{aligned}$$

$$
\begin{aligned}
&= \left| y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)(\psi_{d_k}(x_i)^\top \psi_{d_k}(\theta_\star^{d_k}) + \eta_{d_k}(x_i) + \xi_i) - y^\top \psi_{d_k}(\theta_\star) \right| \\
&= \left| y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)(\eta_{d_k}(x_i) + \xi_i) \right| \\
&\leqslant \left| y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\eta_{d_k}(x_i) \right| + \left| y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\xi_i \right|. \qquad (8.36)
\end{aligned}
$$

We next bound the two terms in Eq. (8.36) separately. For the first term, we have

$$
\begin{aligned}
\left| y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\eta_{d_k}(x_i) \right| &\leqslant \widetilde{\gamma}(d_k) \sum_{i=1}^{N_k} \left| y^\top A_k^{-1} \psi_{d_k}(x_i) \right| \\
&= \widetilde{\gamma}(d_k) \sum_{i=1}^{N_k} \sqrt{\left(y^\top A_k^{-1} \psi_{d_k}(x_i) \right)^2} \\
&\leqslant \widetilde{\gamma}(d_k) \sqrt{N_k \sum_{i=1}^{N_k} \left(y^\top A_k^{-1} \psi_{d_k}(x_i) \right)^2} \qquad (8.37) \\
&= \widetilde{\gamma}(d_k) \sqrt{N_k \sum_{i=1}^{N_k} y^\top A_k^{-1} \psi_{d_k}(x_i) \psi_{d_k}(x_i)^\top A_k^{-1} y} \\
&= \widetilde{\gamma}(d_k) \sqrt{N_k \|y\|^2_{A_k^{-1}}} \\
&\leqslant \widetilde{\gamma}(d_k) \sqrt{(1+\zeta)\iota(\mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k)))} \qquad (8.38) \\
&\leqslant \widetilde{\gamma}(d_k) \sqrt{(1+\zeta)\iota(\mathcal{Y}(\psi_{d_k}(\mathcal{S}_k)))} \qquad (8.39)
\end{aligned}
$$

where Eq. (8.37) comes from Jensen's inequality; Eq. (8.38) comes from the guarantee of rounding in Eq. (8.3); and Eq. (8.39) comes from the fact that $\widehat{\mathcal{S}}_k \subseteq \mathcal{S}_k$.

For the second term in Eq. (8.36), since ξ_is are independent 1-sub-Gaussian random variables, we know that the random variable $y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\xi_i$ has variance proxy $\sqrt{\sum_{i=1}^{N_k} \left(y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i) \right)^2} = \|y\|_{A_k^{-1}}$. Combining the standard

Hoeffding's inequality with a union bound leads to

$$\mathbb{P}\left(\forall y \in \mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k)), \left|y^\top A_k^{-1} \sum_{i=1}^{N_k} \psi_{d_k}(x_i)\xi_i\right| \leqslant \|y\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)}\right) \geqslant 1-\delta_k, \tag{8.40}$$

where we use the fact that $|\mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k))| \leqslant |\widehat{\mathcal{S}}_k|^2/2$ in the union bound.

Putting Eq. (8.38) and Eq. (8.40) together, we have

$$\mathbb{P}(\forall y \in \mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k)), |\langle y, \widehat{\theta}_k - \psi_{d_k}(\theta_\star^{d_k})\rangle| \leqslant \widetilde{\gamma}(d_k)\iota_k + \omega_k(y)) \geqslant 1-\delta_k, \tag{8.41}$$

where $\iota_k := \sqrt{(1+\zeta)\iota(\mathcal{Y}(\psi_{d_k}(\mathcal{S}_k)))}$ and $\omega_k(y) := \|y\|_{A_k^{-1}}\sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)}$.

Step 1.3: Correctness. We prove $z_\star \in \widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$ under the good event analyzed in Eq. (8.41).

Step 1.3.1: $z_\star \in \widehat{\mathcal{S}}_{k+1}$**.** For any $\widehat{z} \in \widehat{\mathcal{S}}_k$ such that $\widehat{z} \neq z_\star$, we have

$$\begin{aligned}
&\langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \widehat{\theta}_k\rangle \\
&\leqslant \langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \psi_{d_k}(\theta_\star^{d_k})\rangle + \gamma(d_k)\iota_k + \omega_k(\psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star)) \\
&= h(\widehat{z}) - \eta_{d_k}(\widehat{z}) - h(z_\star) + \eta_{d_k}(z_\star) + \gamma(d_k)\iota_k + \omega_k(\psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star)) \\
&< (2+\iota_k)\widetilde{\gamma}(d_k) + \omega_k(\psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star)) \\
&\leqslant 2^{-k}/2 + 2^{-k}/2 \qquad (8.42) \\
&= 2^{-k},
\end{aligned}$$

where Eq. (8.42) comes from Proposition 8.22 combined with the fact that $d_k \geqslant d_\star$ (as shown in Step 1.1), and the selection of N_k together with the guarantees in the rounding procedure Eq. (8.3).

Step 1.3.2: $\widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$**.** Consider any $z \in \widehat{\mathcal{S}}_k \cap \mathcal{S}_{k+1}^c$, we know that $\Delta_z \geqslant 2\cdot 2^{-k}$ by definition. Since $z_\star \in \widehat{\mathcal{S}}_k$, we then have

$$\langle \psi_{d_k}(z_\star) - \psi_{d_k}(z), \widehat{\theta}_k\rangle$$

$$
\begin{aligned}
&\geqslant \langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \psi_{d_k}(\theta_\star^{d_k}) \rangle - \gamma(d_k)\iota_k - \omega_k(\psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star)) \\
&= h(z_\star) - \eta_{d_k}(z_\star) - h(z) + \eta_{d_k}(z) - \gamma(d_k)\iota_k - \omega_k(\psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star)) \\
&\geqslant 2 \cdot 2^{-k} - (2 + \iota_k)\widetilde{\gamma}(d_k) - \omega_k(\psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star)) \\
&\geqslant 2 \cdot 2^{-k} - 2^{-k}/2 - 2^{-k}/2 \qquad (8.43) \\
&= 2^{-k},
\end{aligned}
$$

where Eq. (8.43) comes from a similar reasoning as appearing in Eq. (8.42). As a result, we have $z \notin \widehat{\mathcal{S}}_{k+1}$ and $\widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$.

To summarize, we prove the induction at iteration $k+1$, i.e.,

$$\mathbb{P}(\mathcal{E}_{k+1} \mid \cap_{i<k+1}\mathcal{E}_i) \geqslant 1 - \delta_k.$$

Step 2: The error probability. The analysis on the error probability is the same as in the Step 2 in the proof of Lemma 8.7. Let $\mathcal{E} = \cap_{i=1}^{n+1}\mathcal{E}_i$ denote the good event, we then have

$$\mathbb{P}(\mathcal{E}) \geqslant 1 - \delta.$$

□

Proof of Theorem 8.13.

Theorem 8.13. *With probability at least* $1-\delta$, *Algorithm 30 starts to output* 2ε*-optimal arms after* $N = \widetilde{O}(\log_2(1/\varepsilon)\max\{\rho^\star_{d_\star(\varepsilon)}(\varepsilon), r_{d_\star(\varepsilon)}(\zeta)\} + 1/\varepsilon^2)$ *samples, where we hide logarithmic terms besides* $\log_2(1/\varepsilon)$ *in the* $\widetilde{O}$ *notation.*

Proof. The proof is decomposed into four steps: (1) locating good subroutines; (2) guarantees for the validation step; (3) bounding error probability and (4) bounding unverifiable sample complexity. For fixed ε, we use shorthand $d_\star = d_\star(\varepsilon)$ throughout the proof.

Step 1: The good subroutines. Consider $B_\star = \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}$ and $n_\star = \lceil \log_2(2/\varepsilon) \rceil$. For any subroutines invoked with $B_i \geqslant B_\star$ and $n_i \geqslant n_\star$, we know that,

from Lemma 8.23, the output set of arms are those with sub-optimality gap $< \varepsilon$. Let $i_\star = \lceil \log_2(B_\star) \rceil$, $j_\star = \lceil \log_2(n_\star) \rceil$ and $\ell_\star = i_\star + j_\star$. We know that in outer loops $\ell \geqslant \ell_\star$, there must exists at least one subroutine invoked with $B_i = 2^{i_\star} \geqslant B_\star$ and $n_i = 2^{j_\star} \geqslant n_\star$. As a result, $\mathcal{A}_\ell$ contains at least one ε-optimal arm for $\ell \geqslant \ell_\star$.

Step 2: The validation step. For any $x \in \mathcal{A}_\ell$, we use $\widehat{h}(x)$ to denote its sample mean after $\lceil 8 \log(2/\delta_\ell)/\varepsilon^2 \rceil$ samples. With 1-sub-Gaussian noise, a standard Hoeffding's inequality shows that and a union bound gives

$$\mathbb{P}\Big(\forall x \in \mathcal{A}_\ell : |\widehat{h}(x) - h(x)| \geqslant \varepsilon/2 \Big) \leqslant \ell \delta_\ell. \tag{8.44}$$

As a result, a 2ε-optimal arm will be selected with probability at least $1 - \ell\delta_\ell$, as long as at least one ε-optimal arm is contained in $\mathcal{A}_\ell$.

Step 3: Error probability. We consider the good event where all subroutines invoked in Algorithm 25 with $B_i \geqslant B_\star$ and (any) n_i correctly output a set of arms with sub-optimality gap $< f(n_i + 1)$, as shown in Lemma 8.23, together with the confidence bound described in Eq. (8.44) in the validation step. This good event clearly happens with probability at least $1 - \sum_{\ell=1}^{\infty} \sum_{i=1}^{\ell} 2\delta_\ell = 1 - \sum_{\ell=1}^{\infty} \delta/(2\ell^2) > 1 - \delta$, after applying a union bound argument. We upper bound the unverifiable sample complexity under this good event in the following.

Step 4: Unverifiable sample complexity. For any subroutine invoked within outer loop $\ell \leqslant \ell_\star$, we know, from Algorithm 29, that its sample complexity is upper bounded by (note that $|\mathcal{Z}|^2 \geqslant 4$ trivially holds true)

$$\begin{aligned} N_\ell &\leqslant n_i \big(B_i \cdot \big(10 \log(|\mathcal{Z}|^2/\delta_{\ell_\star}) \big) + 1 \big) \\ &\leqslant \gamma_\ell \, 11 \log\big(4|\mathcal{Z}|^2 \ell_\star^3 / \delta \big). \end{aligned}$$

The validation step within any outer loop $\ell \leqslant \ell_\star$ takes at most $\ell \cdot \lceil 8 \log(2/\delta_\ell)/\varepsilon^2 \rceil \leqslant 9 \log(8\ell_\star^3/\delta)\ell_\star/\varepsilon^2$ samples. Thus, the total sample complexity up to the end of outer

loops $\ell \leqslant \ell_\star$ is upper bounded by

$$\begin{aligned}
N &\leqslant \sum_{\ell=1}^{\ell_\star} \left(\ell N_\ell + \ell \cdot \left\lceil 8 \log(2/\delta_\ell)/\varepsilon^2 \right\rceil\right) \\
&\leqslant 11 \log\left(4|\mathcal{Z}|^2 \ell_\star^3/\delta\right) \sum_{\ell=1}^{\ell_\star} \ell 2^\ell + 9 \log\left(8\ell_\star^3/\delta\right) \ell_\star^2/\varepsilon^2 \\
&\leqslant 22 \log\left(4|\mathcal{Z}|^2 \ell_\star^3/\delta\right) \ell_\star 2^{\ell_\star} + 9 \log\left(8\ell_\star^3/\delta\right) \ell_\star^2/\varepsilon^2.
\end{aligned}$$

By definition of $\ell_\star$, we have

$$\ell_\star \leqslant \log_2\left(4 \log_2(4/\varepsilon) \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}\right),$$

and

$$\begin{aligned}
2^{\ell_\star} &= 2^{(i_\star + j_\star)} \\
&\leqslant 4\left(\log_2(2/\varepsilon) + 1\right) \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}, \\
&= 4 \log_2(4/\varepsilon) \max\{64\rho^\star_{d_\star}, r_{d_\star}(\zeta)\}.
\end{aligned}$$

Set $\tau_\star = \log_2(4/\varepsilon) \max\left\{\rho^\star_{d_\star}, r_{d_\star}(\zeta)\right\}$. The unverifiable sample complexity is upper bounded by (we only consider the case when $\varepsilon \leqslant 1$ in simplifying the bound: otherwise there is no need to prove anything since $\max_{x \in \mathcal{X}} \Delta_x \leqslant 2$)

$$\begin{aligned}
N &\leqslant 5632\, \tau_\star \cdot \left(\log_2(\tau_\star) + 8\right) \cdot \log\left(4|\mathcal{Z}|^2 (\log_2(\tau_\star) + 8)^3/\delta\right) \\
&\quad + 9/\varepsilon^2 \cdot \left(\log_2(\tau_\star) + 8\right)^2 \cdot \log\left(8(\log_2(\tau_\star) + 8)^3/\delta\right) \\
&= \widetilde{O}\left(\log_2(1/\varepsilon) \max\{\rho^\star_{d_\star}, r_{d_\star}(\zeta)\} + 1/\varepsilon^2\right),
\end{aligned}$$

where we hide logarithmic terms besides $\log(1/\varepsilon)$ in the $\widetilde{O}$ notation. □

8.9.5.3 Identifying the Optimal Arm under misspecification

When the goal is to identify the optimal arm under misspecification, i.e., by choosing $\varepsilon = \Delta_{\min}$, one can apply Algorithm 25 together with Algorithm 29 as the subroutine

(thus removing the $1/\varepsilon^2$ term in sample complexity). This combination works since, with appropriate choice of B, Algorithm 29 is guaranteed to output a subset of arms $\widehat{\mathcal{S}}_{n+1}$ with optimality gap $< \Delta_{\min}$ when $n \geqslant \log_2(2/\Delta_{\min})$. This implies that $\widehat{\mathcal{S}} = \{z_\star\}$ and thus the one can reuse the selection rule of Algorithm 25 by recommending arms contained in the singleton set. Note that we can work with the general transductive linear bandit setting in this case, i.e., we don't require $\mathcal{Z} \subseteq \mathcal{X}$ anymore.

8.9.5.4 Omitted Proofs for the Fixed Budget Setting with Misspecification

Lemma 8.24 and Its Proof.

Lemma 8.24. *Suppose* $64\rho^\star_{d_\star(\varepsilon)}(\varepsilon) \leqslant B \leqslant 128\rho^\star_{d_\star(\varepsilon)}(\varepsilon)$ *and* $T/n \geqslant r_{d_\star(\varepsilon)}(\zeta) + 1$. *Algorithm 26 outputs an arm* $\widehat{z}_\star$ *such that* $\Delta_{\widehat{z}_\star} < f(n+1)$ *with probability at least*

$$1 - n|\mathcal{Z}|^2 \exp\left(-\frac{T}{2560\, n\, \rho^\star_{d_\star(\varepsilon)}(\varepsilon)}\right).$$

Furthermore, an ε*-optimal arm is output as long as* $n \geqslant \log_2(2/\varepsilon)$.

Proof. The proof is similar to the proof of Lemma 8.9, with main differences in dealing with misspecification. We provide the proof here for completeness. We consider event

$$\mathcal{E}_k = \{z_\star \in \widehat{\mathcal{S}}_k \subseteq \mathcal{S}_k\},$$

and prove through induction that, for $k \leqslant \lceil \log_2(2/\varepsilon) \rceil$,

$$\mathbb{P}(\mathcal{E}_{k+1} \mid \cap_{i \leqslant k} \mathcal{E}_i) \geqslant 1 - \delta_k,$$

where the value of $\{\delta_k\}_{k=0}^{\lceil \log_2(2/\varepsilon) \rceil}$ will be specified in the proof. For $n \geqslant k+1$, we have $\widehat{\mathcal{S}}_n \subseteq \widehat{\mathcal{S}}_{k+1}$ due to the nature of the elimination-styled algorithm, which guarantees outputting an arm such that $\Delta_z < f(n+1)$. We use the notation $d_\star = d_\star(\varepsilon)$ throughout the rest of the proof.

Step 1: The induction. The base case $\{z_\star \in \widehat{\mathcal{S}}_1 \subseteq \mathcal{S}_1\}$ holds with probability 1 by construction (thus, we have $\delta_0 = 0$). Conditioned on events $\cap_{i=1}^{k} \mathcal{E}_i$, we next analyze the event $\mathcal{E}_{k+1}$.

Step 1.1: $d_k \geqslant d_\star$**.** We first notice that $\widetilde{D}$ is selected as the largest integer such that $r_{\widetilde{D}}(\zeta) \leqslant T'$. When $T/n \geqslant r_{d_\star}(\zeta) + 1$, we have $\widetilde{D} \geqslant d_\star$ since $T' \geqslant T/n - 1 \geqslant r_{d_\star}(\zeta)$. We remark here that for whatever $d_k \in [\widetilde{D}]$ selected, we always have $r_{d_\star}(\zeta) \leqslant r_{\widetilde{D}}(\zeta) \leqslant T'$ and can thus safely apply the rounding procedure described in Eq. (8.3).

Since $\widehat{\mathcal{S}}_k \subseteq \mathcal{S}_k$, we also have

$$\begin{aligned} g_k(d_\star) &= 2^{2k} \iota(\mathcal{Y}(\psi_{d_\star}(\widehat{\mathcal{S}}_k))) \\ &\leqslant 2^{2k} \iota(\mathcal{Y}(\psi_{d_\star}(\mathcal{S}_k))) \\ &\leqslant 64 \rho^\star_{d_\star}(\varepsilon) \qquad (8.45) \\ &\leqslant B, \qquad (8.46) \end{aligned}$$

where Eq. (8.45) comes from Lemma 8.17 and Eq. (8.46) comes from the assumption. As a result, we know that $d_k \geqslant d_\star$ since $d_k \in [\widetilde{D}]$ is selected as the largest integer such that $g_k(d_k) \leqslant B$.

Step 1.2: Concentration and error probability. Let $\{x_1, \ldots, x_{T'}\}$ be the arms pulled at iteration k and $\{r_1, \ldots, r_{T'}\}$ be the corresponding rewards. Let $\widehat{\theta}_k = A_k^{-1} b_k \in \mathbb{R}^{d_k}$ where $A_k = \sum_{i=1}^{T'} \psi_{d_k}(x_i)\psi_{d_k}(x_i)^\top$, and $b_k = \sum_{i=1}^{T'} \psi_{d_k}(x_i) b_i$. Since $d_k \geqslant d_\star$ and the model is well-specified, we can write $r_i = \langle \theta_\star, x_i \rangle + \xi_i = \langle \psi_{d_k}(\theta_\star), \psi_{d_k}(x_i) \rangle + \xi_i$, where ξ_i is i.i.d. generated zero-mean Gaussian noise with variance 1. Similarly as analyzed in Eq. (8.41), we have

$$\mathbb{P}\Big(\forall y \in \mathcal{Y}(\psi_{d_k}(\widehat{\mathcal{S}}_k)), |\langle y, \widehat{\theta}_k - \psi_{d_k}(\theta_\star) \rangle| \leqslant \widetilde{\gamma}(d_k)\iota_k + \omega_k(y) \Big) \geqslant 1 - \delta_k, \qquad (8.47)$$

where $\iota_k := \sqrt{(1+\zeta)\iota(\mathcal{Y}(\psi_{d_k}(\mathcal{S}_k)))}$ and $\omega_k(y) := \|y\|_{A_k^{-1}} \sqrt{2 \log\Big(|\widehat{\mathcal{S}}_k|^2 / \delta_k \Big)}$.

By setting $\max_{y \in \Psi_{d_k}(\widehat{\mathcal{S}}_k)} \|y\|_{A_k^{-1}} \sqrt{2\log\left(|\widehat{\mathcal{S}}_k|^2/\delta_k\right)} = 2^{-k}/2$, we have

$$\begin{aligned}
\delta_k &= |\widehat{\mathcal{S}}_k|^2 \exp\left(-\frac{1}{8 \cdot 2^{2k} \max_{y \in \Psi_{d_k}(\widehat{\mathcal{S}}_k)} \|y\|^2_{A_k^{-1}}}\right) \\
&\leqslant |\widehat{\mathcal{S}}_k|^2 \exp\left(-\frac{T'}{8 \cdot 2^{2k} (1+\zeta)\, \iota(\mathcal{Y}(\Psi_{d_k}(\widehat{\mathcal{S}}_k)))}\right) \qquad (8.48) \\
&\leqslant |\mathcal{Z}|^2 \exp\left(-\frac{T}{4096\, n\, \rho^\star_{d_\star}(\varepsilon)}\right), \qquad (8.49)
\end{aligned}$$

where Eq. (8.48) comes from the guarantee of the rounding procedure Eq. (8.3); and Eq. (8.49) comes from combining the following facts: (1) $2^{2k}\, \iota(\mathcal{Y}(\Psi_{d_k}(\widehat{\mathcal{S}}_k))) \leqslant B \leqslant 128\rho^\star_{d_\star}(\varepsilon)$; (2) $T' \geqslant T/n - 1 \geqslant T/2n$ (note that $T/n \geqslant r_{d_\star}(\zeta) + 1 \implies T/n \geqslant 2$ since $r_{d_\star}(\zeta) \geqslant 1$); (3) $\widehat{\mathcal{S}}_k \subseteq \mathcal{Z}$ and (4) consider some $\zeta \leqslant 1$ (ζ only affects constant terms).

Step 1.3: Correctness. We prove $z_\star \in \widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$ under the good event analyzed in Eq. (8.47).

Step 1.3.1: $z_\star \in \widehat{\mathcal{S}}_{k+1}$**.** For any $\widehat{z} \in \widehat{\mathcal{S}}_k$ such that $\widehat{z} \neq z_\star$, we have

$$\begin{aligned}
&\langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \widehat{\theta}_k \rangle \\
&\leqslant \left\langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \psi_{d_k}(\theta^{d_k}_\star) \right\rangle + \widetilde{\gamma}(d_k)\iota_k + 2^{-k}/2 \\
&= h(\widehat{z}) - \eta_{d_k}(\widehat{z}) - h(z_\star) + \eta_{d_k}(z_\star) + \widetilde{\gamma}(d_k)\iota_k + 2^{-k}/2 \\
&< (2+\iota_k)\,\widetilde{\gamma}(d_k) + 2^{-k}/2 \\
&\leqslant 2^{-k}/2 + 2^{-k}/2 \qquad (8.50) \\
&= 2^{-k},
\end{aligned}$$

where Eq. (8.50) comes from comes from Proposition 8.22 combined with the fact that $d_k \geqslant d_\star$ (as shown in Step 1.1). As a result, $z_\star$ remains in $\widehat{\mathcal{S}}_{k+1}$ according to the elimination criteria.

Step 1.3.2: $\widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$**.** Consider any $z \in \widehat{\mathcal{S}}_k \cap \mathcal{S}^c_{k+1}$, we know that $\Delta_z \geqslant 2 \cdot 2^{-k}$

by definition. Since $z_\star \in \widehat{\mathcal{S}}_k$, we then have

$$\begin{aligned}
&\langle \psi_{d_k}(z_\star) - \psi_{d_k}(z), \widehat{\theta}_k \rangle \\
&\geqslant \langle \psi_{d_k}(\widehat{z}) - \psi_{d_k}(z_\star), \psi_{d_k}(\theta_\star^{d_k}) \rangle - \widetilde{\gamma}(d_k)\iota_k - 2^{-k}/2 \\
&= h(z_\star) - \eta_{d_k}(z_\star) - h(z) + \eta_{d_k}(z) - \widetilde{\gamma}(d_k)\iota_k - 2^{-k}/2 \\
&\geqslant 2 \cdot 2^{-k} - (2 + \iota_k)\widetilde{\gamma}(d_k) - 2^{-k}/2 \\
&= 2 \cdot 2^{-k} - \gamma(d_k) - 2^{-k}/2 \\
&\geqslant 2^{-k}, \qquad (8.51)
\end{aligned}$$

where Eq. (8.51) comes from a similar reasoning as appearing in Eq. (8.50). As a result, we have $z \notin \widehat{\mathcal{S}}_{k+1}$ and $\widehat{\mathcal{S}}_{k+1} \subseteq \mathcal{S}_{k+1}$.

To summarize, we prove the induction at iteration $k+1$, i.e.,

$$\mathbb{P}(\mathcal{E}_{k+1} \mid \cap_{i<k+1} \mathcal{E}_i) \geqslant 1 - \delta_k.$$

Step 2: The error probability. This step is exactly the same as the Step 2 in the proof of Lemma 8.9. Let $\mathcal{E} = \cap_{i=1}^{n+1} \mathcal{E}_i$ denote the good event, we then have

$$\mathbb{P}(\mathcal{E}) \geqslant 1 - n|\mathcal{Z}|^2 \exp\left(-\frac{T}{4096\, n\, \rho^\star_{d_\star}(\varepsilon)}\right).$$

□

Proof of Theorem 8.15.

Theorem 8.15. *Suppose $\mathcal{Z} \subseteq \mathcal{X}$. If $T = \widetilde{\Omega}\left(\log_2(1/\varepsilon) \max\left\{\rho^\star_{d_\star(\varepsilon)}(\varepsilon), r_{d_\star(\varepsilon)}(\zeta)\right\}\right)$, then Algorithm 27 outputs an 2ε-optimal arm with error probability at most*

$$\begin{aligned}
&\log_2(4/\varepsilon)|\mathcal{Z}|^2 \exp\left(-\frac{T}{4096\, \log_2(4/\varepsilon)\, \rho^\star_{d_\star(\varepsilon)}(\varepsilon)}\right) \\
&\quad + 2(\log_2 T)^2 \exp\left(-\frac{T}{8(\log_2 T)^2/\varepsilon^2}\right).
\end{aligned}$$

Furthermore, if there exist universal constants such that $\max_{x\in\mathcal{X}}\|\psi_{d_\star(\varepsilon)}(x)\|^2 \leqslant c_1$ *and* $\min_{z\in\mathcal{Z}}\|\psi_{d_\star(\varepsilon)}(z_\star) - \psi_{d_\star(\varepsilon)}(z)\|^2 \geqslant c_2$*, the error probability is upper bounded by*

$$O\Bigg(\max\Big\{\log_2(1/\varepsilon)|\mathcal{Z}|^2, (\log_2 T)^2\Big\} \times \exp\left(-\frac{c_2 T}{\max\big\{\log_2(1/\varepsilon), (\log_2 T)^2\big\} c_1 \rho^\star_{d_\star(\varepsilon)}(\varepsilon)}\right)\Bigg).$$

Proof. The proof follows similar steps as the proof of Theorem 8.10. Although we are dealing with a misspecified model, guarantees derived in Lemma 8.24 is similar to the ones in Lemma 8.9. When $\varepsilon \leqslant \Delta_{\min}$, the proof goes almost exactly the same as the proof of Theorem 8.10 (with $\rho^\star_{d_\star}$ replaced by $\rho^\star_{d_\star(\varepsilon)}(\varepsilon)$), and Algorithm 27 identifies the optimal arm. When $\varepsilon > \Delta_{\min}$, we additionally replace $\Delta_{\min}$ by ε and equally split the 2ε slackness between selection and validation steps. We also slightly modify Lemma 8.19 to an ε-relaxed version (e.g., in the derivation of Eq. (8.12), select a $z' \in \mathcal{Z}$ with sub-optimality gap $\leqslant \varepsilon$ and then replace $\Delta_{\min}$ by ε). □

8.9.6 Other Details for Experiments

We set confidence parameter $\delta = 0.05$ in our experiments, and generate rewards with Gaussian noise $\xi_t \sim \mathcal{N}(0,1)$. We parallelize our simulations on a cluster consists of two Intel® Xeon® Gold 6254 Processors.

Similar to Fiez et al. (2019), we use a Frank-Wolfe type of algorithm (Jaggi, 2013) with constant step-size $\frac{2}{k+2}$ (we use k to denote the iteration counter in the Frank-Wolfe algorithm) to approximately solve optimal designs. We terminate the Frank-Wolfe algorithm when the relative change of the design value is smaller than 0.01 or when 1000 iterations are reached. We use the rounding procedure developed in Pukelsheim (2006) to round continuous designs to discrete allocations (with $\zeta = 1$, also see Fiez et al. (2019) for a detailed discussion on the rounding procedure). In the implementation of Algorithm 25, we set $\gamma_\ell = 4^\ell$, $n_i = 4^i$ and

$B_i = 4^{\ell - i}$, which only affect constant terms in our theoretical guarantees. We use a binary search procedure to select d_k in Algorithm 24.

Other Experiment Results. We consider a problem instance with $\mathcal{X} = \mathcal{Z}$ being 100 randomly selected arms from the D dimensional unit sphere. We set reward function $h(x) = \langle \theta_\star, x \rangle$ with $\theta_\star = [\frac{1}{1^2}, \frac{1}{2^2}, \ldots, \frac{1}{d_\star^2}, 0 \ldots, 0]^\top \in \mathbb{R}^D$. We filter out instances whose smallest sub-optimality gap is smaller than 0.08. We set $d_\star = 5$ and vary the ambient dimension $D \in \{25, 50, 75, 100\}$. As in Section 8.7, we evaluate each algorithm with success rate, (unverifiable) sample complexity and runtime. We run 100 independent random trials for each algorithm. Due to computational burdens, we force-stop both algorithms after 50,000 samples; we also force-stop the Frank-Wolfe algorithm when 500 iterations are reached.

Table 8.3: Comparison of success rate with varying ambient dimension.

D	25	50	75	100
RAGE	100%	100%	98%	95%
Ours	91%	98%	97%	98%

Success rates of both algorithms are shown in Table 8.3, and RAGE shows advantages over our algorithm when D is small. Fig. 8.2 shows the sample complexity of both algorithms: Our algorithm adapts to the true dimension $d_\star$ yet RAGE is heavily affected by the increasing ambient dimension D.

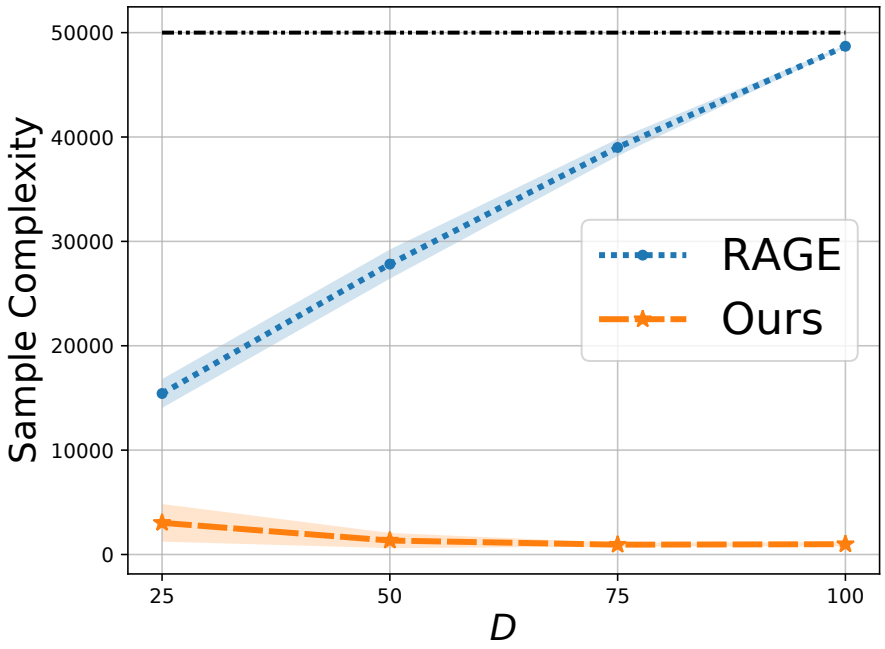


Figure 8.2: Comparison of sample complexity with varying ambient dimension.

The runtime of both algorithms are shown in Table 8.4. RAGE shows clear advantage in runtime and our algorithm suffers from computational overheads of conducting model selection.

Table 8.4: Comparison of runtime with varying ambient dimension.

D	25	50	75	100
RAGE	85.99 s	144.78 s	249.79 s	357.98 s
Ours	287.09 s	339.67 s	489.50 s	678.93 s

We remark that, for the current experiment setups with $d_\star$ and $D \in \{25, 50, 75, 100\}$, our algorithm does not perform well if $\theta_\star$ is chosen to be flat, e.g., $\theta_\star = [\frac{1}{\sqrt{d_\star}}, \ldots, \frac{1}{\sqrt{d_\star}}, 0, \ldots, 0]^\top \in \mathbb{R}^D$. However, we believe that one will eventually see model selection gains if D is chosen to be large enough (and allowing each algorithm takes more samples before force-stopped). One may need to over-

come the computational burdens, e.g., developing practical (or heuristic-based) implementations of our algorithm and RAGE, before running experiments in higher dimensional spaces. We leave large-scale evaluations for future work.

Abbasi-Yadkori, Yasin, Aldo Pacchiano, and My Phan. 2020. Regret balancing for bandit and rl model selection. *arXiv preprint arXiv:2006.05491*.

Abbasi-Yadkori, Yasin, Dávid Pál, and Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. In *Nips*, vol. 11, 2312–2320.

Abbasi-Yadkori, Yasin, David Pal, and Csaba Szepesvari. 2012. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Artificial intelligence and statistics*, 1–9.

Abe, Naoki, Alan W Biermann, and Philip M Long. 2003. Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica* 37(4):263–293.

Abe, Naoki, and Philip M Long. 1999. Associative reinforcement learning using linear probabilistic concepts. In *Icml*, 3–11. Citeseer.

Agarwal, Alekh, Sarah Bird, Markus Cozowicz, Luong Hoang, John Langford, Stephen Lee, Jiaji Li, Dan Melamed, Gal Oshri, Oswaldo Ribas, Siddhartha Sen, and Aleksandrs Slivkins. 2016. Making contextual decisions with low technical debt. *arXiv:1606.03966*.

Agarwal, Alekh, Miroslav Dudík, Satyen Kale, John Langford, and Robert Schapire. 2012. Contextual bandit learning with predictable rewards. In *Artificial intelligence and statistics*, 19–26. PMLR.

Agarwal, Alekh, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. 2014. Taming the monster: A fast and simple algorithm for contextual bandits. In *International conference on machine learning*, 1638–1646. PMLR.

Agarwal, Alekh, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire. 2017. Corralling a band of bandit algorithms. In *Conference on learning theory*, 12–38. PMLR.

Agrawal, Rajeev. 1995. The continuum-armed bandit problem. *SIAM journal on control and optimization* 33(6):1926–1951.

Agrawal, Shipra, and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, 127–135. PMLR.

Alieva, Ayya, Ashok Cutkosky, and Abhimanyu Das. 2021. Robust pure exploration in linear bandits with limited budget. In *International conference on machine learning*, 187–195. PMLR.

Allen-Zhu, Zeyuan, Yuanzhi Li, Aarti Singh, and Yining Wang. 2020. Near-optimal discrete optimization for experimental design: A regret minimization approach. *Mathematical Programming* 1–40.

Anthony, Martin. 2002. Uniform glivenko-cantelli theorems and concentration of measure in the mathematical modelling of learning. *Research Report LSE-CDAM-2002–07*.

Arora, Raman, Teodor V Marinov, and Mehryar Mohri. 2020. Corralling stochastic bandit algorithms. *arXiv preprint arXiv:2006.09255*.

Ash, Jordan, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. 2021. Gone fishing: Neural active learning with fisher embeddings. *Advances in Neural Information Processing Systems* 34.

Ash, Jordan T, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*.

Atwood, Corwin L. 1969. Optimal and efficient designs of experiments. *The Annals of Mathematical Statistics* 1570–1602.

Audibert, Jean-Yves, and Sébastien Bubeck. 2009. Minimax policies for adversarial and stochastic bandits. In *Colt*, 217–226.

Audibert, Jean-Yves, Sébastien Bubeck, and Rémi Munos. 2010. Best arm identification in multi-armed bandits. In *Colt*, 41–53. Citeseer.

Audibert, Jean-Yves, and Alexandre B Tsybakov. 2007. Fast learning rates for plug-in classifiers. *The Annals of statistics* 35(2):608–633.

Auer, Peter. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3(Nov):397–422.

Auer, Peter, Ronald Ortner, and Csaba Szepesvári. 2007. Improved rates for the stochastic continuum-armed bandit problem. In *International conference on computational learning theory*, 454–468. Springer.

Awerbuch, Baruch, and Robert Kleinberg. 2008. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences* 74(1):97–114.

Aziz, Maryam, Jesse Anderton, Emilie Kaufmann, and Javed Aslam. 2018. Pure exploration in infinitely-armed bandit models with fixed-confidence. In *Alt 2018-algorithmic learning theory*.

Aziz, Maryam, Emilie Kaufmann, and Marie-Karelle Riviere. 2021. On multi-armed bandit designs for dose-finding clinical trials. *The Journal of Machine Learning Research* 22(1):686–723.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Balcan, Maria-Florina, Alina Beygelzimer, and John Langford. 2006. Agnostic active learning. In *Proceedings of the 23rd international conference on machine learning*, 65–72.

Balcan, Maria-Florina, Andrei Broder, and Tong Zhang. 2007. Margin based active learning. In *International conference on computational learning theory*, 35–50. Springer.

Bartlett, Peter L, Victor Gabillon, and Michal Valko. 2018. A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. *arXiv preprint arXiv:1810.00997*.

Bartlett, Peter L, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. 2019. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *The Journal of Machine Learning Research* 20(1):2285–2301.

Bergstra, James, and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research* 13(2).

Berner, Christopher, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

Berry, Donald A, Robert W Chen, Alan Zame, David C Heath, and Larry A Shepp. 1997. Bandit problems with infinitely many arms. *The Annals of Statistics* 2103–2116.

Beygelzimer, Alina, Sanjoy Dasgupta, and John Langford. 2009. Importance weighted active learning. In *Proceedings of the 26th annual international conference on machine learning*, 49–56.

Beygelzimer, Alina, Daniel J Hsu, John Langford, and Tong Zhang. 2010. Agnostic active learning without constraints. *Advances in neural information processing systems* 23.

Bhatia, K., K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. 2016. The extreme classification repository: Multi-label datasets and code.

Bhatnagar, Shalabh, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. 2009. Natural Actor–Critic algorithms. *Automatica* 45(11):2471–2482.

Bietti, Alberto, Alekh Agarwal, and John Langford. 2021. A contextual bandit bake-off. *Journal of Machine Learning Research* 22(133):1–49.

Birgé, Lucien, and Pascal Massart. 1997. From model selection to adaptive estimation. In *Festschrift for lucien le cam*, 55–87. Springer.

Boucheron, Stéphane, Olivier Bousquet, and Gábor Lugosi. 2005. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics* 9:323–375.

Bousquet, Olivier, and Nikita Zhivotovskiy. 2021. Fast classification rates without standard margin assumptions. *Information and Inference: A Journal of the IMA* 10(4): 1389–1421.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33:1877–1901.

Bubeck, Sébastien, and Nicolo Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5(1):1–122.

Bubeck, Sébastien, Nicolo Cesa-Bianchi, and Sham M Kakade. 2012. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on learning theory*, 41–1. JMLR Workshop and Conference Proceedings.

Bubeck, Sébastien, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. 2011a. X-armed bandits. *Journal of Machine Learning Research* 12(May):1655–1695.

Bubeck, Sébastien, Gilles Stoltz, and Jia Yuan Yu. 2011b. Lipschitz bandits without the lipschitz constant. In *International conference on algorithmic learning theory*, 144–158. Springer.

Bull, Adam D, et al. 2015. Adaptive-treed bandits. *Bernoulli* 21(4):2289–2307.

Cai, T Tony, Mark G Low, et al. 2005. On adaptive estimation of linear functionals. *The Annals of Statistics* 33(5):2311–2343.

Cai, William, Josh Grossman, Zhiyuan Jerry Lin, Hao Sheng, Johnny Tian-Zheng Wei, Joseph Jay Williams, and Sharad Goel. 2021. Bandit algorithms to personalize educational chatbots. *Machine Learning* 1–30.

Camilleri, Romain, Julian Katz-Samuels, and Kevin Jamieson. 2021. High-dimensional experimental design and kernel bandits. *arXiv preprint arXiv:2105.05806*.

Cao, Tongyi, and Akshay Krishnamurthy. 2019. Disagreement-based combinatorial pure exploration: Sample complexity bounds and an efficient algorithm. In *Conference on learning theory*, 558–588. PMLR.

Carpentier, Alexandra, and Rémi Munos. 2012. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *Artificial intelligence and statistics*, 190–198.

Castro, Rui M, and Robert D Nowak. 2006. Upper and lower error bounds for active learning. In *The 44th annual allerton conference on communication, control and computing*, vol. 2, 1.

———. 2008. Minimax bounds for active learning. *IEEE Transactions on Information Theory* 54(5):2339–2353.

Cesa-Bianchi, Nicolo, and Gábor Lugosi. 2012. Combinatorial bandits. *Journal of Computer and System Sciences* 78(5):1404–1422.

Chatterji, Niladri, Vidya Muthukumar, and Peter Bartlett. 2020. Osom: A simultaneously optimal algorithm for multi-armed and linear contextual bandits. In *International conference on artificial intelligence and statistics*, 1844–1854.

Chaudhuri, Arghya Roy, and Shivaram Kalyanakrishnan. 2018. Quantile-regret minimisation in infinitely many-armed bandits. In *Uai*, 425–434.

Chen, Lijie, Anupam Gupta, Jian Li, Mingda Qiao, and Ruosong Wang. 2017. Nearly optimal sampling algorithms for combinatorial pure exploration. In *Conference on learning theory*, 482–534. PMLR.

Chernozhukov, Victor, Mert Demirer, Greg Lewis, and Vasilis Syrgkanis. 2019. Semi-parametric efficient policy learning with continuous actions. *Advances in Neural Information Processing Systems* 32:15065–15075.

Chow, CK. 1970. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory* 16(1):41–46.

Chu, Wei, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 208–214. JMLR Workshop and Conference Proceedings.

Citovsky, Gui, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. 2021. Batch active learning at scale. *Advances in Neural Information Processing Systems* 34.

Cohen, Michael B, Ben Cousins, Yin Tat Lee, and Xin Yang. 2019. A near-optimal algorithm for approximating the John Ellipsoid. In *Conference on learning theory*, 849–873. PMLR.

Cohn, David, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine learning* 15(2):201–221.

Cortes, Corinna, Giulia DeSalvo, Mehryar Mohri, Ningshan Zhang, and Claudio Gentile. 2019. Active learning with disagreement graphs. In *International conference on machine learning*, 1379–1387. PMLR.

Craven, Peter, and Grace Wahba. 1978. Smoothing noisy data with spline functions. *Numerische mathematik* 31(4):377–403.

Cutkosky, Ashok, and Kwabena Boahen. 2017. Online learning without prior information. *arXiv preprint arXiv:1703.02629*.

Cutkosky, Ashok, Christoph Dann, Abhimanyu Das, Claudio Gentile, Aldo Pacchiano, and Manish Purohit. 2021. Dynamic balancing for model selection in bandits and rl. In *International conference on machine learning*, 2276–2285. PMLR.

Cutkosky, Ashok, Abhimanyu Das, and Manish Purohit. 2020. Upper confidence bounds for combining stochastic bandits. *arXiv preprint arXiv:2012.13115*.

Cutkosky, Ashok, and Francesco Orabona. 2018. Black-box reductions for parameter-free online learning in banach spaces. In *Conference on learning theory*, 1493–1529.

Cybenko, George. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 2(4):303–314.

Dani, Varsha, and Thomas P Hayes. 2006. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *Soda*, vol. 6, 937–943.

Dani, Varsha, Thomas P Hayes, and Sham M Kakade. 2008. Stochastic linear optimization under bandit feedback. *Conference on Learning Theory (COLT)*.

Dasgupta, Sanjoy, Daniel J Hsu, and Claire Monteleoni. 2007. A general agnostic active learning algorithm. *Advances in neural information processing systems* 20.

Dasgupta, Sanjoy, Adam Tauman Kalai, and Adam Tauman. 2009. Analysis of perceptron-based active learning. *Journal of Machine Learning Research* 10(2).

Degenne, Rémy, and Wouter M Koolen. 2019. Pure exploration with multiple correct answers. In *Advances in neural information processing systems*, 14564–14573.

Degenne, Rémy, Pierre Ménard, Xuedong Shang, and Michal Valko. 2020. Gamification of pure exploration for linear bandits. In *International conference on machine learning*, 2432–2442. PMLR.

Dekel, Ofer, Claudio Gentile, and Karthik Sridharan. 2012. Selective sampling and active learning from single and multiple teachers. *The Journal of Machine Learning Research* 13(1):2655–2697.

Deshpande, Yash, and Andrea Montanari. 2012. Linear bandits in high dimension and recommendation systems. In *2012 50th annual allerton conference on communication, control, and computing (allerton)*, 1750–1754. IEEE.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Naacl-hlt (1)*.

Dudik, Miroslav, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. 2011. Efficient optimal learning for contextual bandits. In *Proceedings of the twenty-seventh conference on uncertainty in artificial intelligence*, 169–178.

Edmonds, Jack. 1965. Paths, trees, and flowers. *Canadian Journal of mathematics* 17: 449–467.

Emam, Zeyad Ali Sami, Hong-Min Chu, Ping-Yeh Chiang, Wojciech Czaja, Richard Leapman, Micah Goldblum, and Tom Goldstein. 2021. Active learning at the imagenet scale. *arXiv preprint arXiv:2111.12880*.

Even-Dar, Eyal, Shie Mannor, and Yishay Mansour. 2006. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research* 7(Jun):1079–1105.

Fiez, Tanner, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. 2019. Sequential experimental design for transductive linear bandits. *Advances in neural information processing systems* 32.

Foster, Dylan, Alekh Agarwal, Miroslav Dudík, Haipeng Luo, and Robert Schapire. 2018. Practical contextual bandits with regression oracles. In *International conference on machine learning*, 1539–1548. PMLR.

Foster, Dylan, and Alexander Rakhlin. 2020. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In *International conference on machine learning*, 3199–3210. PMLR.

Foster, Dylan, Alexander Rakhlin, David Simchi-Levi, and Yunzong Xu. 2021a. Instance-dependent complexity of contextual bandits and reinforcement learning:

A disagreement-based perspective. In *Conference on learning theory*, 2059–2059. PMLR.

Foster, Dylan J, Claudio Gentile, Mehryar Mohri, and Julian Zimmert. 2020a. Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems* 33:11478–11489.

Foster, Dylan J, Sham M Kakade, Jian Qian, and Alexander Rakhlin. 2021b. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*.

Foster, Dylan J, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. 2017. Parameter-free online learning via model selection. In *Advances in neural information processing systems*, 6020–6030.

Foster, Dylan J, and Akshay Krishnamurthy. 2021. Efficient first-order contextual bandits: Prediction, allocation, and triangular discrimination. *Advances in Neural Information Processing Systems* 34.

Foster, Dylan J, Akshay Krishnamurthy, and Haipeng Luo. 2019. Model selection for contextual bandits. *arXiv preprint arXiv:1906.00531*.

———. 2020b. Open problem: Model selection for contextual bandits. In *Conference on learning theory*, 3842–3846. PMLR.

Foster, Dylan J, Alexander Rakhlin, David Simchi-Levi, and Yunzong Xu. 2020c. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. *arXiv preprint arXiv:2010.03104*.

Freedman, David A. 1975. On tail probabilities for martingales. *the Annals of Probability* 100–118.

Friedman, Eric. 2009. Active learning for smooth problems. In *Colt*. Citeseer.

Gaillard, Pierre, and Sébastien Gerchinovitz. 2015. A chaining algorithm for online nonparametric regression. In *Conference on learning theory*, 764–796. PMLR.

Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International conference on machine learning*, 1183–1192. PMLR.

Garivier, Aurélien, Hédi Hadiji, Pierre Menard, and Gilles Stoltz. 2018. Kl-ucb-switch: optimal regret bounds for stochastic bandits from both a distribution-dependent and a distribution-free viewpoints. *arXiv preprint arXiv:1805.05071*.

Ghalme, Ganesh, Swapnil Dhamal, Shweta Jain, Sujit Gujar, and Y Narahari. 2020. Ballooning multi-armed bandits. *arXiv preprint arXiv:2001.10055*.

Ghosh, Avishek, Abishek Sankararaman, and Kannan Ramchandran. 2020. Problem-complexity adaptive model selection for stochastic linear bandits. *arXiv preprint arXiv:2006.02612*.

Grill, Jean-Bastien, Michal Valko, and Rémi Munos. 2015. Black-box optimization of noisy functions with unknown smoothness. In *Advances in neural information processing systems*, 667–675.

Grötschel, Martin, László Lovász, and Alexander Schrijver. 2012. *Geometric algorithms and combinatorial optimization*, vol. 2. Springer Science & Business Media.

Guruswami, Venkatesan, and Prasad Raghavendra. 2009. Hardness of learning halfspaces with noise. *SIAM Journal on Computing* 39(2):742–765.

Hadiji, Hédi. 2019. Polynomial cost of adaptation for X-armed bandits. *Advances in Neural Information Processing Systems* 32.

Han, Yanjun, Zhengqing Zhou, Zhengyuan Zhou, Jose Blanchet, Peter W Glynn, and Yinyu Ye. 2020. Sequential batch learning in finite-action linear contextual bandits. *arXiv preprint arXiv:2004.06321*.

Hanneke, Steve. 2007. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on machine learning*, 353–360.

———. 2014. Theory of active learning. *Foundations and Trends in Machine Learning* 7(2-3).

Hao, Botao, Tor Lattimore, and Mengdi Wang. 2020. High-dimensional sparse linear bandits. *arXiv preprint arXiv:2011.04020*.

Haussler, David. 1989. Decision theoretic generalizations of the pac model for neural net and other learning applications.

———. 1992. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation* 100(1):78–150.

———. 1995. Sphere packing numbers for subsets of the boolean n-cube with bounded vapnik-chervonenkis dimension. *Journal of Combinatorial Theory, Series A* 69(2):217–232.

Hazan, Elad, Amit Agarwal, and Satyen Kale. 2007. Logarithmic regret algorithms for online convex optimization. *Machine Learning* 69(2-3):169–192.

Hazan, Elad, and Zohar Karnin. 2016. Volumetric spanners: An efficient exploration basis for learning. *The Journal of Machine Learning Research* 17(1):4062–4095.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 770–778.

Heinonen, Juha. 2005. *Lectures on lipschitz analysis*. 100, University of Jyväskylä.

Herbei, Radu, and Marten H Wegkamp. 2006. Classification with reject option. *The Canadian Journal of Statistics/La Revue Canadienne de Statistique* 709–721.

Hoffman, Matthew, Bobak Shahriari, and Nando Freitas. 2014. On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Artificial intelligence and statistics*, 365–374. PMLR.

Hornik, Kurt. 1991. Approximation capabilities of multilayer feedforward networks. *Neural networks* 4(2):251–257.

Hsu, Daniel Joseph. 2010. Algorithms for active learning. Ph.D. thesis, UC San Diego.

Huang, Tzu-Kuo, Alekh Agarwal, Daniel J Hsu, John Langford, and Robert E Schapire. 2015. Efficient and parsimonious agnostic active learning. *Advances in Neural Information Processing Systems* 28.

Ito, Shinji, Daisuke Hatano, Hanna Sumita, Kei Takemura, Takuro Fukunaga, Naonori Kakimura, and Ken-Ichi Kawarabayashi. 2019. Oracle-efficient algorithms for online linear optimization with bandit feedback. *Advances in Neural Information Processing Systems* 32:10590–10599.

Jaggi, Martin. 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, 427–435. PMLR.

John, F. 1948. Extremum problems with inequalities as subsidiary conditions. *R. Courant Anniversary Volume* 187–204.

Jun, Kwang-Sung, Aniruddha Bhargava, Robert Nowak, and Rebecca Willett. 2017. Scalable generalized linear bandits: Online computation and hashing. *Advances in Neural Information Processing Systems* 30.

Kääriäinen, Matti. 2006. Active learning in the non-realizable case. In *International conference on algorithmic learning theory*, 63–77. Springer.

Kakade, Sham M, Varun Kanade, Ohad Shamir, and Adam Kalai. 2011. Efficient learning of generalized linear and single index models with isotonic regression. *Advances in Neural Information Processing Systems* 24.

Karzand, Mina, and Robert D Nowak. 2020. Maximin active learning in overparameterized model classes. *IEEE Journal on Selected Areas in Information Theory* 1(1):167–177.

Kassraie, Parnian, and Andreas Krause. 2022. Neural contextual bandits without regret. In *International conference on artificial intelligence and statistics*, 240–278. PMLR.

Katz-Samuels, Julian, Lalit Jain, Kevin G Jamieson, et al. 2020. An empirical process approach to the union bound: Practical algorithms for combinatorial and linear bandits. *Advances in Neural Information Processing Systems* 33.

Katz-Samuels, Julian, and Kevin Jamieson. 2019. The true sample complexity of identifying good arms. *arXiv preprint arXiv:1906.06594*.

———. 2020. The true sample complexity of identifying good arms. In *International conference on artificial intelligence and statistics*, 1781–1791. PMLR.

Katz-Samuels, Julian, Jifan Zhang, Lalit Jain, and Kevin Jamieson. 2021. Improved algorithms for agnostic pool-based active classification. In *International conference on machine learning*, 5334–5344. PMLR.

Kaufmann, Emilie, Olivier Cappé, and Aurélien Garivier. 2016. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research* 17(1):1–42.

Kiefer, Jack, and Jacob Wolfowitz. 1960. The equivalence of two extremum problems. *Canadian Journal of Mathematics* 12:363–366.

Kleinberg, Robert. 2004. Nearly tight bounds for the continuum-armed bandit problem. *Advances in Neural Information Processing Systems* 17:697–704.

Kleinberg, Robert, Aleksandrs Slivkins, and Eli Upfal. 2008. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual acm symposium on theory of computing*, 681–690.

Kleinberg, Robert D. 2005. Nearly tight bounds for the continuum-armed bandit problem. In *Advances in neural information processing systems*, 697–704.

Koltchinskii, Vladimir. 2010. Rademacher complexities and bounding the excess risk in active learning. *The Journal of Machine Learning Research* 11:2457–2485.

Koolen, Wouter M, and Tim Van Erven. 2015. Second-order quantile methods for experts and combinatorial games. In *Conference on learning theory*, 1155–1175.

Kothawade, Suraj, Nathan Beck, Krishnateja Killamsetty, and Rishabh Iyer. 2021. Similar: Submodular information measures based active learning in realistic scenarios. *Advances in Neural Information Processing Systems* 34.

Kpotufe, Samory, Gan Yuan, and Yunfan Zhao. 2021. Nuances in margin conditions determine gains in active learning. *arXiv preprint arXiv:2110.08418*.

Krishnamurthy, Akshay, Alekh Agarwal, Tzu-Kuo Huang, Hal Daumé III, and John Langford. 2017. Active learning for cost-sensitive classification. In *International conference on machine learning*, 1915–1924. PMLR.

———. 2019. Active learning for cost-sensitive classification. *Journal of Machine Learning Research* 20:1–50.

Krishnamurthy, Akshay, John Langford, Aleksandrs Slivkins, and Chicheng Zhang. 2020. Contextual bandits with continuous actions: Smoothing, zooming, and adapting. *Journal of Machine Learning Research* 21(137):1–45.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25.

Langford, John, and Tong Zhang. 2007. The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in neural information processing systems* 20(1):96–1.

Lattimore, Tor. 2015. The pareto regret frontier for bandits. *arXiv preprint arXiv:1511.00048*.

———. 2020. Improved regret for zeroth-order adversarial bandit convex optimisation. *Mathematical Statistics and Learning* 2(3):311–334.

Lattimore, Tor, and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

Lattimore, Tor, Csaba Szepesvari, and Gellert Weisz. 2020. Learning with good feature representations in bandits and in rl with a generative model. In *International conference on machine learning*, 5662–5670. PMLR.

Lebret, Rémi, and Ronan Collobert. 2014. Word embeddings through Hellinger PCA. In *Proceedings of the 14th conference of the european chapter of the association for computational linguistics*, 482–490.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521(7553):436–444.

Lepskii, OV. 1991. On a problem of adaptive estimation in gaussian white noise. *Theory of Probability & Its Applications* 35(3):454–466.

Li, Gene, Pritish Kamath, Dylan J Foster, and Nathan Srebro. 2021. Eluder dimension and generalized rank. *arXiv preprint arXiv:2104.06970*.

Li, Lihong, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on world wide web*, 661–670.

Locatelli, Andrea, and Alexandra Carpentier. 2018. Adaptivity to smoothness in x-armed bandits. In *Conference on learning theory*, 1463–1492.

Locatelli, Andrea, Alexandra Carpentier, and Samory Kpotufe. 2017. Adaptivity to noise parameters in nonparametric active learning. In *Proceedings of the 2017 conference on learning theory, pmlr*.

———. 2018. An adaptive strategy for active learning with smooth decision boundary. In *Algorithmic learning theory*, 547–571. PMLR.

Lu, Jianfeng, Zuowei Shen, Haizhao Yang, and Shijun Zhang. 2021. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis* 53(5): 5465–5506.

Luo, Haipeng, and Robert E Schapire. 2015. Achieving all with no parameters: Adanormalhedge. In *Conference on learning theory*, 1286–1304.

Mahabadi, Sepideh, Piotr Indyk, Shayan Oveis Gharan, and Alireza Rezaei. 2019. Composable core-sets for determinant maximization: A simple near-optimal algorithm. In *International conference on machine learning*, 4254–4263. PMLR.

Majzoubi, Maryam, Chicheng Zhang, Rajan Chari, Akshay Krishnamurthy, John Langford, and Aleksandrs Slivkins. 2020. Efficient contextual bandits with continuous actions. *Advances in Neural Information Processing Systems* 33:349–360.

Mannor, Shie, and John N Tsitsiklis. 2004. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research* 5(Jun): 623–648.

Marinov, Teodor Vanislavov, and Julian Zimmert. 2021. The pareto frontier of model selection for general contextual bandits. *Advances in Neural Information Processing Systems* 34.

Massart, Pascal, and Élodie Nédélec. 2006. Risk bounds for statistical learning. *The Annals of Statistics* 34(5):2326–2366.

McMahan, Brendan, and Jacob Abernethy. 2013. Minimax optimal algorithms for unconstrained linear optimization. *Advances in Neural Information Processing Systems* 26:2724–2732.

McMahan, H Brendan, and Avrim Blum. 2004. Online geometric optimization in the bandit setting against an adaptive adversary. In *International conference on computational learning theory*, 109–123. Springer.

Meyer, Carl D. 2000. *Matrix analysis and applied linear algebra*, vol. 71. Siam.

Minsker, Stanislav. 2012. Plug-in approach to active learning. *Journal of Machine Learning Research* 13(1).

Ongie, Greg, Rebecca Willett, Daniel Soudry, and Nathan Srebro. 2020. A function space view of bounded norm infinite width relu nets: The multivariate case. In *International conference on learning representations*.

Orabona, Francesco. 2014. Simultaneous model selection and optimization through parameter-free stochastic learning. *Advances in Neural Information Processing Systems* 27:1116–1124.

Orabona, Francesco, and Dávid Pál. 2016. Coin betting and parameter-free online learning. *Advances in Neural Information Processing Systems* 29:577–585.

Oswal, Urvashi, Aniruddha Bhargava, and Robert Nowak. 2020. Linear bandits with feature feedback. In *Aaai*, 5331–5338.

Pacchiano, Aldo, Christoph Dann, Claudio Gentile, and Peter Bartlett. 2020a. Regret bound balancing and elimination for model selection in bandits and rl. *arXiv preprint arXiv:2012.13045*.

Pacchiano, Aldo, My Phan, Yasin Abbasi-Yadkori, Anup Rao, Julian Zimmert, Tor Lattimore, and Csaba Szepesvari. 2020b. Model selection in contextual stochastic bandit problems. *arXiv preprint arXiv:2003.01704*.

Pan, Feiyang, Qingpeng Cai, Pingzhong Tang, Fuzhen Zhuang, and Qing He. 2019. Policy gradients for contextual recommendations. In *The world wide web conference*, 1421–1431.

Papini, Matteo, Andrea Tirinzoni, Marcello Restelli, Alessandro Lazaric, and Matteo Pirotta. 2021. Leveraging good representations in linear contextual bandits. *arXiv preprint arXiv:2104.03781*.

Parhi, Rahul, and Robert D Nowak. 2021. Banach space representer theorems for neural networks and ridge splines. *J. Mach. Learn. Res.* 22(43):1–40.

———. 2022a. Near-minimax optimal estimation with shallow relu neural networks. *IEEE Transactions on Information Theory*.

———. 2022b. What kinds of functions do deep neural networks learn? insights from variational spline theory. *SIAM Journal on Mathematics of Data Science* 4(2): 464–489.

Pollard, D. 1984. *Convergence of stochastic processes*. David Pollard.

Puchkin, Nikita, and Nikita Zhivotovskiy. 2021. Exponential savings in agnostic active learning through abstention. *arXiv preprint arXiv:2102.00451*.

Pukelsheim, Friedrich. 2006. *Optimal design of experiments*. SIAM.

Rahimi, Ali, Benjamin Recht, et al. 2007. Random features for large-scale kernel machines. In *Nips*, vol. 3, 5. Citeseer.

Réda, Clémence, Emilie Kaufmann, and Andrée Delahaye-Duriez. 2020. Machine learning applications in drug development. *Computational and structural biotechnology journal* 18:241–252.

Reimers, Nils, and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing*. Association for Computational Linguistics.

Ren, Pengzhen, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM Computing Surveys (CSUR)* 54(9):1–40.

Ruan, Yufei, Jiaqi Yang, and Yuan Zhou. 2021. Linear bandits with limited adaptivity and learning distributional optimal design. In *Proceedings of the 53rd annual acm sigact symposium on theory of computing*, 74–87.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115:211–252.

Russo, Daniel, and Benjamin Van Roy. 2013. Eluder dimension and the sample complexity of optimistic exploration. In *Nips*, 2256–2264. Citeseer.

———. 2018. Satisficing in time-sensitive bandit learning. *arXiv preprint arXiv:1803.02855*.

Ryzhov, Ilya O, Warren B Powell, and Peter I Frazier. 2012. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research* 60(1): 180–195.

Sahni, Sartaj. 1974. Computationally related problems. *SIAM Journal on computing* 3(4):262–279.

Schrijver, Alexander. 1998. *Theory of linear and integer programming*. John Wiley & Sons.

Sen, Rajat, Alexander Rakhlin, Lexing Ying, Rahul Kidambi, Dean Foster, Daniel N Hill, and Inderjit S Dhillon. 2021. Top-k extreme contextual bandits with arm hierarchy. In *International conference on machine learning*, 9422–9433. PMLR.

Sener, Ozan, and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *International conference on learning representations*.

Settles, Burr. 2009. Active learning literature survey.

Shalev-Shwartz, Shai, and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shang, Xuedong, Emilie Kaufmann, and Michal Valko. 2019. General parallel optimization a without metric. In *Algorithmic learning theory*, 762–788.

Shao, Jun. 1993. Linear model selection by cross-validation. *Journal of the American statistical Association* 88(422):486–494.

Shawe-Taylor, John, Peter L Bartlett, Robert C Williamson, and Martin Anthony. 1998. Structural risk minimization over data-dependent hierarchies. *IEEE transactions on Information Theory* 44(5):1926–1940.

Shekhar, Shubhanshu, Mohammad Ghavamzadeh, and Tara Javidi. 2021. Active learning for classification with abstention. *IEEE Journal on Selected Areas in Information Theory* 2(2):705–719.

Sherman, Jack, and Winifred J Morrison. 1950. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics* 21(1):124–127.

Shrivastava, Anshumali, and Ping Li. 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). *Advances in neural information processing systems* 27.

Silver, David, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature* 529(7587):484–489.

Simchi-Levi, David, and Yunzong Xu. 2020. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Available at SSRN 3562765*.

———. 2021. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*.

Singh, Sameer, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Tech. Rep. UM-CS-2012-015, University of Massachusetts, Amherst.

Soare, Marta, Alessandro Lazaric, and Rémi Munos. 2014. Best-arm identification in linear bandits. *Advances in Neural Information Processing Systems* 27.

Stone, M. 1978. Cross-validation: A review. *Statistics: A Journal of Theoretical and Applied Statistics* 9(1):127–139.

Stone, Mervyn. 1974. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B* (*Methodological*) 36(2): 111–133.

Summa, Marco Di, Friedrich Eisenbrand, Yuri Faenza, and Carsten Moldenhauer. 2014. On largest volume simplices and sub-determinants. In *Proceedings of the twenty-sixth annual acm-siam symposium on discrete algorithms*, 315–323. SIAM.

Tanczos, Ervin, Robert Nowak, and Bob Mankoff. 2017. A kl-lucb algorithm for large-scale crowdsourcing. *Advances in Neural Information Processing Systems* 30.

Tao, Chao, Saúl Blanco, and Yuan Zhou. 2018. Best arm identification in linear bandits with linear dimension dependency. In *International conference on machine learning*, 4877–4886.

Tewari, Ambuj, and Susan A Murphy. 2017. From Ads to interventions: Contextual bandits in mobile health. In *Mobile health*, 495–517. Springer.

Teytaud, Olivier, Sylvain Gelly, and Michèle Sebag. 2007. Anytime many-armed bandits. In *CAP07*. Grenoble, France.

Tong, Simon, and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research* 2(Nov): 45–66.

Tsybakov, Alexander B. 2004. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics* 32(1):135–166.

Unser, Michael. 2022. Ridges, neural networks, and the radon transform. *arXiv preprint arXiv:2203.02543*.

Valiant, Leslie G. 1984. A theory of the learnable. *Communications of the ACM* 27(11):1134–1142.

Valko, Michal, Alexandra Carpentier, and Rémi Munos. 2013. Stochastic simultaneous optimistic optimization. In *International conference on machine learning*, 19–27.

Vapnik, Vladimir, and Alexey Chervonenkis. 1974. Theory of pattern recognition.

Vapnik, Vladimir N. 1995. The nature of statistical learning theory.

Vapnik, VN, and A Ya Chervonenkis. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications* 16(2):264.

Vasnetsov, Andrey. 2018. Oneshot-wikilinks. https://www.kaggle.com/generall/oneshotwikilinks.

Vovk, Vladimir. 1998. A game of prediction with expert advice. *Journal of Computer and System Sciences* 56(2):153–173.

Wagenmaker, Andrew, Julian Katz-Samuels, and Kevin Jamieson. 2021. Experimental design for regret minimization in linear bandits. In *International conference on artificial intelligence and statistics*, 3088–3096.

Wainwright, Martin J. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*, vol. 48. Cambridge University Press.

Wang, Liwei. 2011. Smoothness, disagreement coefficient, and the label complexity of agnostic active learning. *Journal of Machine Learning Research* 12(7).

Wang, Yizao, Jean-Yves Audibert, and Rémi Munos. 2008. Algorithms for infinitely many-armed bandits. *Advances in Neural Information Processing Systems* 21.

———. 2009. Algorithms for infinitely many-armed bandits. In *Advances in neural information processing systems*, 1729–1736.

Wang, Zhilei, Pranjal Awasthi, Christoph Dann, Ayush Sekhari, and Claudio Gentile. 2021. Neural active learning with performance guarantees. *Advances in Neural Information Processing Systems* 34.

Williams, Ronald J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3):229–256.

Xu, Liyuan, Junya Honda, and Masashi Sugiyama. 2018. A fully adaptive algorithm for pure exploration in linear bandits. In *International conference on artificial intelligence and statistics*, 843–851.

Xu, Pan, Zheng Wen, Handong Zhao, and Quanquan Gu. 2020. Neural contextual bandits with deep representation and shallow exploration. *arXiv preprint arXiv:2012.01780*.

Xu, Yunbei, and Assaf Zeevi. 2020. Upper counterfactual confidence bounds: a new optimism principle for contextual bandits. *arXiv preprint arXiv:2007.07876*.

Yang, Junwen, and Vincent YF Tan. 2021. Towards minimax optimal best arm identification in linear bandits. *arXiv preprint arXiv:2105.13017*.

Yang, Shuo, Tongzheng Ren, Sanjay Shakkottai, Eric Price, Inderjit S Dhillon, and Sujay Sanghavi. 2021. Linear bandit algorithms with sublinear time complexity. *arXiv preprint arXiv:2103.02729*.

Yao, Andrew Chi-Chin. 1977. Probabilistic computations: Toward a unified measure of complexity. In *18th annual symposium on foundations of computer science (sfcs 1977)*, 222–227. IEEE Computer Society.

Yarotsky, Dmitry. 2017. Error bounds for approximations with deep relu networks. *Neural Networks* 94:103–114.

———. 2018. Optimal approximation of continuous functions by very deep relu networks. In *Conference on learning theory*, 639–649. PMLR.

Zhang, Chicheng, and Kamalika Chaudhuri. 2014. Beyond disagreement-based agnostic active learning. *Advances in Neural Information Processing Systems* 27.

Zhang, Tong. 2021. Feel-good thompson sampling for contextual bandits and reinforcement learning. *arXiv preprint arXiv:2110.00871*.

Zhang, Weitong, Dongruo Zhou, Lihong Li, and Quanquan Gu. 2021. Neural Thompson sampling. In *International conference on learning representation (iclr)*.

Zhou, Dongruo, Lihong Li, and Quanquan Gu. 2020. Neural contextual bandits with ucb-based exploration. In *International conference on machine learning*, 11492–11502. PMLR.

Zhou, Yuan, Xi Chen, and Jian Li. 2014. Optimal pac multiple arm identification with applications to crowdsourcing. In *International conference on machine learning*, 217–225. PMLR.

Zhu, Yinglun, Dylan J Foster, John Langford, and Paul Mineiro. 2022a. Contextual bandits with large action spaces: Made practical. In *International conference on machine learning*, 27428–27453. PMLR.

Zhu, Yinglun, Julian Katz-Samuels, and Robert D Nowak. 2022b. Near instance optimal model selection for pure exploration linear bandits. In *International conference on artificial intelligence and statistics*, 6735–6769. PMLR.

Zhu, Yinglun, and Paul Mineiro. 2022. Contextual bandits with smooth regret: Efficient learning in continuous action spaces. In *International conference on machine learning*, 27574–27590. PMLR.

Zhu, Yinglun, and Robert D Nowak. 2020. On regret with multiple best arms. *Advances in Neural Information Processing Systems* 33:9050–9060.

———. 2022a. Active learning with neural networks: Insights from nonparametric statistics. *Advances in Neural Information Processing Systems* 35.

———. 2022b. Efficient active learning with abstention. *Advances in Neural Information Processing Systems*.

———. 2022c. Pareto optimal model selection in linear bandits. In *International conference on artificial intelligence and statistics*, 6793–6813. PMLR.

Zhu, Yinglun, Dongruo Zhou, Ruoxi Jiang, Quanquan Gu, Rebecca Willett, and Robert Nowak. 2021. Pure exploration in kernel and neural bandits. *Advances in neural information processing systems* 34:11618–11630.