University of California
Santa Barbara

Adaptive Sequential Decision Making: Bandit Optimization and Active Learning

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

Doctor of Philosophy
in
Computer Science

by

Chong Liu

Committee in charge:

Professor Yu-Xiang Wang, Chair
Professor Xifeng Yan
Professor Anton Van der Ven

September 2023

The Dissertation of Chong Liu is approved.

Professor Xifeng Yan

Professor Anton Van der Ven

Professor Yu-Xiang Wang, Committee Chair

June 2023

Adaptive Sequential Decision Making: Bandit Optimization and Active Learning

by

Chong Liu

To my family.

Acknowledgements

This thesis could not have been possible without my great advisor Yu-Xiang Wang at UCSB Computer Scinece (CS). Yu-Xiang may be one of the best advisors that a PhD student can hope for, both academically and personally. He is a true computer scientist and statistician and I learned a lot from him. He is not only technically solid but also very helpful. It's not easy for me to learn how to do theoretical research work at the very beginning of my PhD but he is always very knowledgeable and patient. It helped me a lot when I was in my junior PhD years. Also, since Day 1 of my PhD journey, Yu-Xiang gave me total freedom to pursue the research directions and I worked on crowdsourcing, differential privacy, and active learning. Since active learning, I found my true interests lie in adaptive sequential decision making, including active learning, Bayesian optimization, and bandits, and I believe decision making is one of the key parts of modern Machine Learning (ML). This is also why the thesis exists. In addition, Yu-Xiang generously supports me to attend many conferences, workshops, and seminars where I can not only keep up with the latest research but also build deep connections with my research community and finally make my career decision to be part of it. I deeply thank Yu-Xiang for his advice, expertise, patience, and support for the last five years and I'll cherish them forever.

This thesis is also advised and supported by Xifeng Yan from UCSB CS and Anton Van der Ven from UCSB Materials. I'm grateful that Xifeng provided me with insightful empirical perspectives in addition to my theoretical research and they are really helpful for me to find problems with strong real-world motivations. Also, Xifeng gave me a lot of career advice. With the collaboration with Anton, my research area expands beyond ML into scientific domains and I'm really excited about solving practical problems in new material design.

During my PhD journey, I have a wonderful team of co-authors Ming Yin, Chuanhao Li, Ilija Bogunovic, Dan Qiao, Kamalika Chaudhuri, Yuqing Zhu, Derick Ober, Anirudh Raju Natarajan, Yuyin Sun, Nan Qiao, Cheng-Hao Kuo, Lu Xia, Jiajia Luo, and Ke Zhang. I appreciate their expertise and contributions to our papers.

I really enjoyed technical discussions with Yuandong Tian, Zhongxiang Dai, Mladen Kolar, Sen Na, Masashi Sugiyama, Christos Thrampoulidis, Jingfeng Zhang, Woody Zhu, Zack Lipton, Jing Lei, Chenlu Ye, Elijah Cole, Gengchen Mai, and Samuel Stanton at conferences, on campuses, or via Zoom. Among them, I especially thank Masashi and Christos for inviting me to speak at RIKEN and UBC.

Besides research, Peter Cappello and Ziad Matni showed me how to be a great teacher when I was a teaching assistant with their courses. In addition, Trinabh Gupta and Subash Suri taught great graduate courses where I liked their teaching and learned a lot beyond my research areas.

Since 2022 Fall, my job hunting received advice and support from Lirong Xia, Willie Neiswanger, Alex Smola, Mengdi Wang, Jie Shen, Xiangnan Kong, Liping Liu, Yisong Yue, Kyunghyun Cho, William Wang, Qinghua Ding, Guo Yu, Mengyang Gu, Jacob Gardner, Aaditya Ramdas, Petko Bogdanov, Ming-Ching Chang, Yiming Ying, Yuxin Chen, and Rebecca Willett. I'm glad all your support finally pays off!

Starting from scratch, I'm the lead organizer of the NeurIPS 2023 Workshop on New Frontiers of AI for Drug Discovery and Development (AI4D3 2023). It could not be accepted without the help from Quanquan Gu, Ti-Chiun Chang, Wei Wang, Michelle M Li, Natasha Tagasovska, and Anima Anandkumar. I also thank Marinka Zitnik, Haoda Fu, and Jian Tang for accepting my invitations to speak at the workshop. I look forward to seeing you all in New Orleans in December!

I want to thank UCSB CS staff members Benji Dunson, Karen van Gool, and Bella Cardoso for their highly efficient communication and two of them have been promoted!

I'll always remember my labmates and friends Jianyu Xu, Dheeraj Baby, Xuandong Zhao, Rachel Redberg, Kaiqi Zhang, Esha Singh, Erchi Wang, Xiyou Zhou, Zhiyu Chen, Hong Wang, Wanrong Zhu, An Yan, and Hanwen Zha. I really enjoyed our time together, either in dicussion, traveling, or throwing a BBQ party!

I want to thank my basketball friends Kaikai, Ken, Vincent, Eason, Bowen, Enbo, Andrew, Jeremy, Joseph, Johnny, Jesse, and Ben. I enjoyed every beautiful sunset while we were playing at Girsh park, Goleta. I especially thank my friend Yifan Lu, who is not only my basketball coach but also my golf coach. Unfortunately I'm not a big fan of fishing otherwise he will be my great fishing coach too for sure!

I also would like to thank my flight instructor Juliet Davis who helped me build my wings such that I can fly independently as a private pilot. Taking friends for a night flight from Santa Barbara to Los Angeles was absolutely an once-in-the-lifetime experience!

In the end, I deeply thank my family for their love and unconditional support throughout my PhD journey. I love them too.

Curriculum Vitæ

Chong Liu

Education

2023	Ph.D. in Computer Science, University of California, Santa Barbara, California, USA.
2018	M.S. in Computer Science, Nanjing University, Nanjing, China.
2015	B.E. in Electrical Engineering, Northwestern Polytechnical University, Xi'an, China.

Publications

1. **No-Regret Linear Bandits beyond Realizability.**
   Chong Liu, Ming Yin, and Yu-Xiang Wang.
   The 39th Conference on Uncertainty in Artificial Intelligence (**UAI**-2023), Pittsburgh, PA, 2023, pp. 1294-1303.
2. **Global Optimization with Parametric Function Approximation.**
   Chong Liu and Yu-Xiang Wang.
   The 40th International Conference on Machine Learning (**ICML**-2023), Honolulu, HI, 2023, pp. 22113-22136.
3. **Human-in-the-Loop Video Semantic Segmentation Auto-Annotation.**
   Nan Qiao, Yuyin Sun, Chong Liu, Lu Xia, Jiajia Luo, Ke Zhang, and Cheng-Hao Kuo.
   The 10th IEEE/CVF Winter Conference on Applications of Computer Vision (**WACV**-2023), Waikoloa, HI, 2023, pp. 5881-5891.
4. **Doubly Robust Crowdsourcing.**
   Chong Liu and Yu-Xiang Wang.
   Journal of Artificial Intelligence Research (**JAIR**), 73:209-229, 2022.
5. **Revisiting Model-Agnostic Private Learning: Faster Rates and Active Learning.**
   Chong Liu, Yuqing Zhu, Kamalika Chaudhuri, and Yu-Xiang Wang.
   Journal of Machine Learning Research (**JMLR**), 22(262):1-44, 2021.
6. **Revisiting Model-Agnostic Private Learning: Faster Rates and Active Learning.**
   Chong Liu, Yuqing Zhu, Kamalika Chaudhuri, and Yu-Xiang Wang.
   The 24th International Conference on Artificial Intelligence and Statistics (**AISTATS**-2021), San Diego, CA, 2021, pp. 838-846.

Abstract

Adaptive Sequential Decision Making: Bandit Optimization and Active Learning

by

Chong Liu

Deep neural networks usually have many hyperparameters that need to be tuned. Modern material design problems usually require material scientists to sequentially select processing parameters and conduct experiments to observe material performances. To save privacy cost, the learning system needs to carefully choose queries to answer under the differential privacy framework. To train a robot under video guidance, engineers need to carefully choose video samples for training. However, in all cases, people cannot observe performances of unselected actions and the experimental cost can be huge. These two challenges hinder efficient neural network training, new material design, privacy protection, and robot training and call for actions. In this thesis, I present my research on optimization, bandits, and active learning under the adaptive sequential decision making framework. My algorithms are able to solve black box function optimization without the curse of dimensionality, achieve no regret under the function class misspecification, reduce privacy cost under the differential privacy framework, and significantly reduce video sample complexity for robot training. All of them come with theoretical or empirical analysis.

Contents

Chapter 1

Introduction

Modern material design problems usually require material scientists to sequentially select processing parameters ahead of time and then conduct expensive experiments to observe material performances. However, material scientists cannot observe performances of unselected parameters. The experimental cost can also be huge if parameters are not selected well such that material performances fail to meet a given criterion.

To train a good neural network, computer scientists need to carefully choose a set of hyperparameters of the neural network. One of the most popular ways to tune hyperparameters is to use grid search but it cannot work well if there are many hyperparameters. Even worse, every time a certain set of hyperparameters is chosen for training on a validation set, the training time is long and computational cost is huge if the neural network is deep in layers.

These two challenges, no output of unselected parameters and huge experimental cost, hinder new material design and neural network training and call for actions.

From the point of view of machine learning, this kind of problems falls into the framework of adaptive sequential decision making with bandit feedback, shown below. During total T rounds, at each round t, decision makers take action x_t and observe

feedback $f(x_t)$ of selected action only, then they use all information collected before to learn about the environment and decide where to take actions afterwards to achieve a certain given goal.

Adaptive Decision Making Framework

1: **for** $t = 1, ..., T$ **do**
2: Take action $x_t \in \mathbb{R}^d$ based on $\{x_1, f(x_1), ..., x_{t-1}, f(x_{t-1})\}$.
3: Observe feedback $f(x_t) \in \mathbb{R}$.
4: **end for**

For example, in material design problem, x_t is the set of processing parameters, such as temperature, pressure, solution concentration, and time, and $f(x_t)$ is the material performance. In neural network hyperparameter tuning task, x_t is the set of hyperparameters, such as learning rate, batch size, number of iterations, and $f(x_t)$ is the validation accuracy on validation set.

Systematically studying the framework and applying results to practical problems are extremely exciting and have significant real-world impacts. Motivated by but not limited to these two real-world problems, this thesis presents my research within the adaptive sequential decision making framework and has two parts. In theory part, global black-box optimization and misspecified linear bandits are studied. In application part, active learning is used to protect privacy under differential privacy framework and to save annotation cost for video semantic segmentation tasks. All four chapters later are introduced briefly as follows.

Global black-box function optimization. In many material design problems, material performance can be modeled as a black-box function of processing parameters where the function needs to be either maximized or minimized. For example, in [1], material scientists want to synthesize ceramic TiO_2 thin films using microwave radiation where

the film property is a black-box non-convex function of temperature, solution concentration, pressure, and processing time. Here scientists need to sequentially select a set of processing parameters such that an ideal film can be synthesized in the end.

Existing methods, such as Bayesian Optimization (BO), have been deployed in solving such problems and show promising performances. [2] used BO to find structural parameters that maximize the energy absorption under compression and successfully reduced number of experiments from $1,800$ (required by grid search) to only 100, saving a lot of experimental cost. However, by assuming the black-box objective function is drawn from Gaussian process, BO suffers from the curse of dimensionality and works poorly when the input dimension d is larger than 20. Developing efficient global optimization algorithms is an interesting research question.

Chapter 2 solves the global black-box optimization problem with parametric function approximation and doesn't suffer from the curse of dimensionality. Theoretically, under the realizable assumption and geometric conditions on parameter class, the new GO-UCB algorithm can achieve global optima with a cumulative regret of $\tilde{O}(\sqrt{T})$ where T is number of total rounds. The regret bound is input dimension-free which means GO-UCB works well in high dimensionality as long as a good parametric function is used for approximation. At the core of GO-UCB is a carefully designed parameter uncertainty set that allows optimistic exploration. Real-world experiments also show that GO-UCB works better than classical BO approaches in high dimensional cases, even if the model is misspecified.

No-regret misspecified linear bandits. Chapter 3 technically considers the same problem as Chapter 2 but in a more challenging setting, misspecified bandits. Note theoretical results in Chapter 2 hold if the realizability assumption holds. Realizability assumption says that the approximation function class always contains the true underlying function, which is almost impossible in real-world applications because one can hardly

capture a function that is assumed to be unknown and black-boxed. Therefore, studying misspecified bandits (without realizability assumption) is the main goal of Chapter 3.

To get started, Chapter 3 only studies the misspecified linear bandits. Existing work usually assumes uniform misspecification but under this condition, most algorithms can only achieve the $\tilde{O}(\sqrt{T} + \epsilon T)$ regret and unfortunately the ϵT term is unavoidable. To overcome it, this chapter proposes the first misspecification condition under which classical LinUCB algorithm [16] achieves $\tilde{O}(\sqrt{T})$ regret, a.k.a. no regret in theory. Starting from linear bandits, there are many potential future directions awaiting to be pursued, such as misspecified generalized linear bandits and misspecified kernelized bandits.

Disagreement-based active learning for privacy protection. In classification tasks, compared with traditional supervised learning, active learning allows the learner to actively select data points to query their labels, thus total labeling cost can be saved and even classifier performance may improve. Motivated by success of active learning, Chapter 4 revisits the model-agnostic private learning framework. The key idea is that by answering fewer queries selected by active learning, privacy loss can be saved thus stronger privacy protection can be obtained without hurting classification performance. Based on this idea, the new PATE-ASQ algorithm is proposed and it is proved to satisfy differential privacy guarantee and achieves almost the same learning bound as non-private supervised learning algorithm. Later PATE-ASQ is also implemented and works well in practice.

Active sample selection for video semantic segmentation. Instead of classification task, Chapter 5 studies the video semantic segmentation annotation problem where video samples are selected for human annotator to get annotated. Traditional methods suffer from either high annotation cost or low performance. With the help of active learning, the new human-in-the-loop algorithm designed in this chapter is able to achieve high semantic segmentation performance while saving the annotation cost at the same time.

After introduction, four chapters are written based on four published papers [3, 10, 5, 6] so each has its own problem setup, related work, results, and notation systems. For quick access, readers may directly jump to the preferred chapter to see how adaptive sequential decision making works for that specific problem. Although all chapters seem relatively independent to each other, they all sit under the adaptive sequential decision making framework and need to quantify the uncertainty and take sequential actions. The differences between these problem settings and algorithms are also interesting to read. In the end, Chapter 6 concludes this thesis.

Chapter 2

Global Black-Box Optimization

This chapter considers the problem of global optimization with noisy zeroth order oracles — a well-motivated problem useful for various applications ranging from hyper-parameter tuning for deep learning to new material design. Existing work relies on Gaussian processes or other non-parametric family, which suffers from the curse of dimensionality. In this chapter, we propose a new algorithm GO-UCB that leverages a parametric family of functions (e.g., neural networks) instead. Under a realizable assumption and a few other mild geometric conditions, we show that GO-UCB achieves a cumulative regret of $\tilde{O}(\sqrt{T})$ where T is the time horizon. At the core of GO-UCB is a carefully designed uncertainty set over parameters based on gradients that allows optimistic exploration. Synthetic and real-world experiments illustrate GO-UCB works better than popular Bayesian optimization approaches, even if the model is misspecified.

2.1 Introduction

We consider the problem of finding a global optimal solution to the following optimization problem

$$\max_{x \in \mathcal{X}} f(x),$$

where $f : \mathcal{X} \to \mathbb{R}$ is an unknown non-convex function that is not necessarily differentiable in x.

This problem is well-motivated by many real-world applications. For example, the accuracy of a trained neural network on a validation set is complex non-convex function of a set of hyper-parameters (e.g., learning rate, momentum, weight decay, dropout, depth, width, choice of activation functions ...) that one needs to maximize [11]. Also in material design, researchers want to synthesize ceramic materials, e.g., titanium dioxide (TiO_2) thin films, using microwave radiation [1] where the film property is a non-convex function of parameters including temperature, solution concentration, pressure, and processing time. Efficiently solving such non-convex optimization problems could significantly reduce energy cost.

We assume having access to only noisy function evaluations, i.e., at round t, we select a point x_t and receive a noisy function value y_t,

$$y_t = f(x_t) + \eta_t, \tag{2.1}$$

where η_t for $t = 1, ..., T$ are *independent*, *zero-mean*, *σ-sub-Gaussian* noise. This is known as the *noisy zeroth-order oracle* setting in optimization literature. Let f^* be the optimal function value, following the tradition of Bayesian optimization (see e.g., [12] for a review), throughout this chapter, we use *cumulative regret* as the evaluation criterion,

defined as

$$R_T = \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} f^* - f(x_t),$$

where r_t is called instantaneous regret at round t. An algorithm $\mathcal{A}$ is said to be a no-regret algorithm if $\lim_{T\to\infty} R_T(\mathcal{A})/T = 0$.

Generally speaking, solving a global non-convex optimization is NP-hard [13] and we need additional assumptions to efficiently proceed. Bayesian optimization usually assumes the objective function f is drawn from a Gaussian process prior. [14] proposed the GP-UCB approach, which iteratively queries the argmax of an upper confidence bound of the current posterior belief, before updating the posterior belief using the new data point. However, Gaussian process relies on kernels, e.g., squared error kernel or Matérn kernel, which suffer from the curse of dimensionality. A folklore rule-of-thumb is that GP-UCB becomes unwieldy when the dimension is larger than 10.

A naive approach is to passively query T data points uniformly at random, estimate f by $\hat{f}$ using supervised learning, then return the maximizer of the plug-in estimator $\hat{x} = \text{argmax}_{x\in\mathcal{X}} \hat{f}(x)$. This may side-step the curse-of-dimensionality depending on which supervised learning model we use. The drawback of this passive query model is that it does not consider the structure of the function nor does it quickly "zoom-in" to the region of the space that is nearly optimal. In contrast, an active query model allows the algorithm to iteratively interact with the function. At round t, the model collects information from all previous rounds $1, ..., t-1$ and decides where to query next.

GO-UCB Algorithm. In this chapter, we develop an algorithm that allows Bayesian optimization-style active queries to work for general supervised learning-based function approximation. We assume that the supervised learning model $f_w : \mathcal{X} \to \mathbb{R}$ is differentiable w.r.t. its d_w-dimensional parameter vector $w \in \mathcal{W} \subset \mathbb{R}^{d_w}$ and that the function

class $\mathcal{F} = \{f_w | w \in \mathcal{W}\}$ is flexible enough such that the true objective function $f = f_{w^*}$ for some $w^* \in \mathcal{W}$, i.e., $\mathcal{F}$ is *realizable*. Our algorithm — *Global Optimization via Upper Confidence Bound* (GO-UCB) — has two phases:

The *GO-UCB* Framework:

- Phase I: Uniformly explore n data points.
- Phase II: Optimistically explore T data points.

The goal of Phase I to sufficiently explore the function and make sure the estimated parameter $\hat{w}_0$ is close enough to true parameter w^* such that exploration in Phase II are efficient. To solve the estimation problem, we rely on a regression oracle that is able to return an estimated $\hat{w}_0$ after n observations. In details, after Phase I we have a dataset $\{(x_j, y_j)\}_{j=1}^n$, then

$$\hat{w}_0 \leftarrow \operatorname*{argmin}_{w \in \mathcal{W}} \sum_{j=1}^{n} (f_w(x_j) - y_j)^2. \tag{2.2}$$

This problem is known as a *non-linear least square* problem. It is computationally hard in the worst-case, but many algorithms are known (e.g., SGD, Gauss-Newton, Levenberg-Marquardt) to effectively solve this problem in practice. Our theoretical analysis of $\hat{w}_0$ uses techniques from [15]. See Section 2.5.1 for details.

In Phase II, exploration is conducted following the principle of "Optimism in the Face of Uncertainty", i.e., the parameter is optimized within an uncertainty region that always contains the true parameter w^*. Existing work in bandit algorithms provides techniques that work when f_w is a linear function [16] or a generalized linear function [17], but no solution to general differentiable function is known. At the core of our GO-UCB is a carefully designed uncertainty ball Ball_t over parameters based on gradients, which allows techniques from the linear bandit [16] to be adapted for the non-linear case. In detail,

the ball is defined to be centered at $\hat{w}_t$ — the solution to a regularized online regression problem after $t-1$ rounds of observations. And the radius of the ball is measured by the covariance matrix of the gradient vectors of all previous rounds. We prove that w^* is always trapped within the ball with high probability.

Contributions. In summary, our main contributions are:

1. We initiate the study of global optimization problem with parametric function approximation and proposed a new optimistic exploration algorithm — GO-UCB.

2. Assuming *realizability* and other mild geometric conditions, we prove that GO-UCB converges to the global optima with cumulative regret at the order of $\tilde{O}(\sqrt{T})$ where T is the time horizon.

3. GO-UCB does not suffer from the curse of dimensionality like Gaussian processes-based Bayesian optimization methods. The unknown objective function f can be high-dimensional, non-convex, non-differentiable, and even discontinuous in its input domain.

4. Synthetic test function and real-world hyperparameter tuning experiments show that GO-UCB works better than all compared Bayesian optimization methods in both realizable and misspecified settings.

Technical novelties. The design of GO-UCB algorithm builds upon the work of [16] and [18], but requires substantial technical novelties as we handle a generic nonlinear parametric function approximation. Specifically:

1. LinUCB analysis (e.g., self-normalized Martingale concentration, elliptical potential lemmas [16, 18]) is not applicable for nonlinear function approximation, but we showed that they can be adapted for this purpose if we can *localize* the learner to a neighborhood of w^*.

2. We identify a new set of structural assumptions under which we can localize the learner sufficiently with only $O(\sqrt{T})$ rounds of pure exploration.

3. Showing that w^* remains inside the parameter uncertainty ball $\text{Ball}_t, \forall t \in [T]$ is challenging. We solve this problem by setting regularization centered at the initialization parameter $\hat{w}_0$ and presenting novel inductive proof of a lemma showing $\forall t \in [T], \hat{w}_t$ converges to w^* in ℓ_2-distance at the same rate.

These new techniques could be of independent interest.

2.2 Related Work

Global non-convex optimization is an important problem that can be found in a lot of research communities and real-world applications, e.g., optimization [19, 20], machine learning [21, 22], hyperparameter tuning [23], neural architecture search [24, 25], and material discovery [26].

One of the most prominent approaches to this problem is Bayesian Optimization (BO) [27], in which the objective function is usually modeled by a Gaussian Process (GP) [28], so that the uncertainty can be updated under the Bayesian formalism. Among the many notable algorithms in GP-based BO [14, 29, 30, 31, 32, 33], GP-UCB [14] is the closest to this chapter because our algorithm also selects data points in a UCB (upper confidence bound) style but the construction of the UCB in this chapter is different since we are not working with GPs. [34] proves lower bounds on regret for noisy Gaussian process bandit optimization. GPs are highly flexible and can approximate any smooth functions, but such flexibility comes at a price to play — curse of dimensionality. Most BO algorithms do not work well when $d > 10$. Notable exceptions include the work of [35, 36, 37, 38, 39] who designed more specialized BO algorithms for high-dimensional tasks.

Besides BO with GPs, other nonparametric families were considered for global optimization tasks, but they, too, suffer from the curse of dimensionality. We refer readers to [40] and the references therein.

While most BO methods use GP as surrogate models, there are other BO methods that use alternative function classes such as neural networks [41, 42]. These methods are different from us in that they use different ways to fit the neural networks and a Monte Carlo sampling approach to decide where to explore next. Empirically, it was reported that they do not outperform advanced GP-based methods that use trust regions [37].

Our problem is also connected to the bandits literature [43, 44, 45, 46]. The global optimization problem can be written as a nonlinear bandits problem in which queried points are actions and the function evaluations are rewards. However, no bandits algorithms can simultaneously handle an infinite action space and a generic nonlinear reward function. Here "generic" means the reward function is much more general than a linear or generalized linear function [46]. To the best of our knowledge, we are the first to address the infinite-armed bandit problems with a general differentiable value function (albeit with some additional assumptions).

A recent line of work studied bandits and global optimization with neural function approximation [47, 48, 49]. The main difference from us is that these results still rely on Gaussian processes with a Neural Tangent Kernel in their analysis, thus intrinsically linear. Their regret bounds also require the width of the neural network to be much larger than the number of samples to be sublinear. In contrast, our results apply to general nonlinear function approximations and do not require overparameterization.

2.3 Preliminaries

2.3.1 Notations

We use $[n]$ to denote the set $\{1, 2, ..., n\}$. The algorithm queries n points in Phase I and T points in Phase II. Let $\mathcal{X} \subset \mathbb{R}^{d_x}$ and $\mathcal{Y} \subset \mathbb{R}$ denote the domain and range of f, and $\mathcal{W} \subset [0,1]^{d_w}$ denote the parameter space of a family of functions $\mathcal{F} := \{f_w : \mathcal{X} \to \mathcal{Y} | w \in \mathcal{W}\}$. For convenience, we denote the bivariate function $f_w(x)$ by $f_x(w)$ when w is the variable of interest. $\nabla f_x(w)$ and $\nabla^2 f_x(w)$ denote the gradient and Hessian of function f w.r.t. w. $L(w) := \mathbb{E}_{x\sim\mathcal{U}}(f_x(w) - f_x(w^*))^2$ denotes the (expected) risk function where $\mathcal{U}$ is uniform distribution. For a vector x, its ℓ_p norm is denoted by $\|x\|_p = (\sum_{i=1}^{d} |x_i|^p)^{1/p}$ for $1 \le p < \infty$ and its ℓ_∞ norm is denoted by $\|x\|_\infty = \max_{i\in[d_x]} |x_i|$. For a matrix A, its operator norm is denoted by $\|A\|_{\text{op}}$. For a vector x and a square matrix A, define $\|x\|_A^2 = x^\top Ax$. Throughout this chapter, we use standard big O notation that hide universal constants; and to improve the readability, we use $\tilde{O}$ to hide all logarithmic factors as well as all polynomial factors in problem-specific parameters except $d_w, 1/\mu, T$. For reader's easy reference, we list all symbols and notations in Table 2.1.

2.3.2 Assumptions

Here we list main assumptions that we will work with throughout this chapter. The first assumption says that we have access to a differentiable function family that contains the unknown objective function.

Assumption 2.3.1 (Realizability) *There exists $w^* \in \mathcal{W}$ such that the unknown objective function $f = f_{w^*}$. Also, assume $\mathcal{W} \subset [0,1]^{d_w}$. This is w.l.o.g. for any compact $\mathcal{W}$.*

Realizable parameter class is a common assumption in literature [50, 51, 44], usually the

Table 2.1: Symbols and notations.

Symbol	Definition	Description
$\|A\|_{\text{op}}$		operator norm
Ball_t	eq. (2.6)	parameter uncertainty region at round t
β_t	eq. (2.7)	parameter uncertainty region radius at round t
μ		local strong convexity parameter
c		local self-concordance parameter
d_x		domain dimension
d_w		parameter dimension
δ		failure probability
ε		covering number discretization distance
η	σ-sub-Gaussian	observation noise
$f_w(x)$		objective function at x parameterized by w
$f_x(w)$		objective function at w parameterized by x
$\nabla f_x(w)$		1st order derivative w.r.t. w parameterized by x
$\nabla^2 f_x(w)$		2nd order derivative w.r.t. w parameterized by x
F		function range constant bound
γ, τ		growth condition parameters
ι, ι', ι''		logarithmic terms
$L(w)$	$\mathbb{E}[(f_x(w) - f_x(w^*))^2]$	expected loss function
λ		regularization parameter
n		time horizon in Phase I
$[n]$	$\{1, 2, ..., n\}$	integer set of size n
r_t	$f_{w^*}(x^*) - f_{w^*}(x_t)$	instantaneous regret at round t
R_T	$\sum_{t=1}^{T} r_t$	cumulative regret after round T
Σ_t	eq. (2.3)	covariance matrix at round t
T		time horizon in Phase II
$\mathcal{U}$		uniform distribution
w	$w \in \mathcal{W}$	function parameter
w^*	$w^* \in \mathcal{W}$	true parameter
$\hat{w}_0$		oracle-estimated parameter after Phase I
$\hat{w}_t$	eq. (2.5)	updated parameter at round t
$\mathcal{W}$	$\mathcal{W} \subseteq [0, 1]^{d_w}$	parameter space
x	$x \in \mathcal{X}$	data point
x^*		optimal data point
$\|x\|_p$	$(\sum_{i=1}^{d} \|x_i\|^p)^{1/p}$	ℓ_p norm
$\|x\|_\infty$	$\max_{i \in [d]} \|x_i\|$	ℓ_∞ norm
$\|x\|_A$	$\sqrt{x^\top A x}$	distance defined by square matrix A
$\mathcal{X}$	$\mathcal{X} \subseteq \mathbb{R}^{d_x}$	function domain
$\mathcal{Y}$	$\mathcal{Y} = [-F, F]$	function range

starting point of a line of research for a new problem because one doesn't need to worry about extra regret incurred by misspecified parameter. Although in this chapter we only theoretically study the realizable parameter class, our GO-UCB algorithm empirically works well in misspecified tasks too.

The second assumption is on properties of the function approximation.

Assumption 2.3.2 (Bounded, differentiable and smooth function approximation) *There exist constants* $F, C_g, C_h > 0$ *such that* $\forall x \in \mathcal{X}, \forall w \in \mathcal{W}$, *it holds that* $|f_x(w)| \leq F$,

$$\|\nabla f_x(w)\|_2 \leq C_g, \quad \text{and} \quad \|\nabla^2 f_x(w)\|_{\text{op}} \leq C_h.$$

This assumption imposes mild regularity conditions on the smoothness of the function with respect to its parameter vector w.

The third assumption is on the expected loss function over the uniform distribution (or any other exploration distribution) in the Phase I of GO-UCB.

Assumption 2.3.3 (Geometric conditions on the loss function) $L(w) = \mathbb{E}_{x\sim\mathcal{U}}(f_x(w) - f_x(w^*))^2$ *satisfies* (τ, γ)-growth condition *or* μ-local strong convexity *at* w^*, *i.e.,* $\forall w \in \mathcal{W}$,

$$\min\left\{\frac{\mu}{2}\|w - w^*\|_2^2, \frac{\tau}{2}\|w - w^*\|_2^\gamma\right\} \leq L(w) - L(w^*),$$

for constants $\mu, \tau > 0, \mu < d_w$ *and* $0 < \gamma < 2$. *Also,* $L(w)$ *satisfies a* c-local self-concordance *assumption at* w^*, *i.e., for all* w *s.t.* $\|w - w^*\|_{\nabla^2 L(w^*)} \leq c$,

$$(1 - c)^2 \cdot \nabla^2 L(w^*) \preceq \nabla^2 L(w) \preceq (1 - c)^{-2} \cdot \nabla^2 L(w^*).$$

We also assume $c \leq 0.5$ *for convenience. This is without loss of generality because if the condition holds for* $c > 0.5$, *then the condition for* $c \leq 0.5$ *is automatically satisfied.*

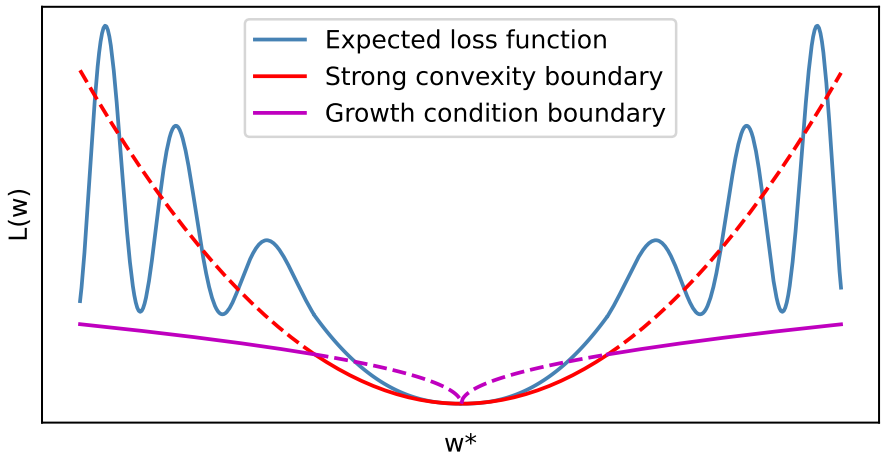


Figure 2.1: Example of a highly non-convex $L(w)$ satisfying Assumption 2.3.3. Solid lines denote the actual lower bound by taking min over strong convexity and growth condition. $L(w)$ is strongly convex near w^* but can be highly non-convex away from w^*.

This assumption has three main components: (global) growth condition, local strong convexity, and local self-concordance.

The global growth condition says that f_w with parameters far away from w^* cannot approximate f well over the distribution $\mathcal{U}$. The local strong convexity assumption requires the local neighborhood near w^* to have quadratic growth.

These two conditions are strictly weaker than global strong convexity because it does not require convexity except in a local neighborhood near the global optimal w^*, i.e., $\{w | \|w - w^*\|_2 \leq (\tau/\mu)^{\frac{1}{2-\gamma}}\}$ and it does not limit the number of spurious local minima, as the global γ-growth condition only gives a mild lower bound as w moves away from w^*. See Figure 2.1 for an example. Our results work even if γ is a small constant < 1.

Self-concordance originates from a clean analysis of Newton's method [52]. See Example 4 of [53] for a list of examples satisfying self-concordance. A localized version of self-concordance is needed in our problem for technical reasons, but again it is only required within a small ball of radius c near w^* for the expected loss under $\mathcal{U}$. Our results

work even if c vanishes at $O(T^{-1/4})$.

To avoid any confusion, the three assumptions we made above are only about the expected loss function w.r.t. uniform distribution $\mathcal{U}$ as a function of w, rather than objective function $f_{w^*}(x)$. The problem to be optimized can still be arbitrarily complex in terms of $\mathcal{X}$, e.g., high-dimensional and non-continuous functions. As an example, in Gaussian process-based Bayesian optimization approaches, $f_{w^*}(x)$ belongs to a reproducing kernel Hilbert space, but its loss function is globally convex in its "infinite dimensional" parameter w. Also, we no longer need this assumption in Phase II.

Additional notations. For convenience, we define $\zeta > 0$ such that $\|\nabla^2 L(w^*)\|_{\text{op}} \leq \zeta$. The existence of a finite ζ is implied by Assumption 2.3.2 and it suffices to take $\zeta = 2C_g^2$ because $\nabla^2 L(w^*) = \mathbb{E}_{x\sim\mathcal{U}}[2\nabla f_x(w^*)\nabla f_x(w^*)^\top]$.

2.4 Main Results

In Section 2.4.1, we state our Global Optimization with Upper Confidence Bound (GO-UCB) algorithm and explain key design points of it. Then in Section 2.4.2, we prove that its cumulative regret bound is at the rate of $\tilde{O}(\sqrt{T})$.

2.4.1 Algorithm

Our GO-UCB algorithm, shown in Algorithm 1, has two phases. Phase I does uniform exploration in n rounds and Phase II does optimistic exploration in T rounds. In Step 1 of Phase I, n is chosen to be large enough such that the objective function can be sufficiently explored. Step 2-3 are doing uniform sampling. In Step 5, we call regression oracle to estimate $\hat{w}_0$ given all observations in Phase I as in eq. (2.2). Adapted from [15], we prove the convergence rate of $\|\hat{w}_0 - w^*\|_2$ is at the rate of $\tilde{O}(1/\sqrt{n})$. See Theorem 2.5.2 for details.

The key challenge of Phase II of GO-UCB is to design an acquisition function to select $x_t, \forall t \in [T]$. Since we are using parametric function to approximate the objective function, we heavily rely on a feasible parameter uncertainty region $\text{Ball}_t, \forall t \in [T]$, which should always contain the true parameter w^* throughout the process. The shape of Ball_t is measured by the covariance matrix Σ_t, defined as

$$\Sigma_t = \lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top. \tag{2.3}$$

Note i is indexing over both x and w, which means that as time t goes from 0 to T, the update to Σ_t is always rank one. It allows us to bound the change of Σ_t from $t = 0$ to T.

Ball_t is centered at $\hat{w}_t$, the newly estimated parameter at round t. In Step 2, we

Algorithm 1 GO-UCB

Input: Time horizon T, uniform exploration phase length n, uniform distribution $\mathcal{U}$, regression oracle Oracle, regularization weight λ, confidence sequence β_t for $t = 1, 2, ..., T$.

Phase I (Uniform exploration)

1: **for** $j = 1, ..., n$ **do**
2: Sample $x_j \sim \mathcal{U}(\mathcal{X})$.
3: Observe $y_j = f(x_j) + \eta_j$.
4: **end for**
5: Estimate $\hat{w}_0 \leftarrow \text{Oracle}(x_1, y_1, ..., x_n, y_n)$.

Phase II (Optimistic exploration)

1: **for** $t = 1, ..., T$ **do**
2: Update Σ_t by eq. (2.3) with the input λ.
3: Update $\hat{w}_t$ by eq. (2.5) with the input λ.
4: Update Ball_t by eq. (2.6) with the input β_t.
5: Select $x_t = \text{argmax}_{x \in \mathcal{X}} \max_{w \in \text{Ball}_t} f_x(w)$.
6: Observe $y_t = f(x_t) + \eta_t$.
7: **end for**

Output: $\hat{x} \sim \mathcal{U}(\{x_1, ..., x_T\})$.

update the estimated $\hat{w}_t$ by solving the following optimization problem:

$$\hat{w}_t = \operatorname*{argmin}_w \frac{\lambda}{2}\|w - \hat{w}_0\|_2^2 + \frac{1}{2}\sum_{i=0}^{t-1}((w - \hat{w}_i)^\top \nabla f_{x_i}(\hat{w}_i) + f_{x_i}(\hat{w}_i) - y_i)^2. \tag{2.4}$$

The optimization problem is an online regularized least square problem involving gradients from all previous rounds, i.e., $\nabla f_{x_i}(\hat{w}_i), \forall i \in [T]$. The intuition behind it is that we use gradients to approximate the function since we are dealing with generic objective function. We set the regularization w.r.t. $\hat{w}_0$ rather than 0 because from regression oracle we know how close is $\hat{w}_0$ to w^*. By setting the gradient of objective function in eq. (2.4) to be 0, the closed form solution of $\hat{w}_t$ is

$$\hat{w}_t = \Sigma_t^{-1}\left(\sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)(\nabla f_{x_i}(\hat{w}_i)^\top \hat{w}_i + y_i - f_{x_i}(\hat{w}_i))\right) + \lambda\Sigma_t^{-1}\hat{w}_0. \tag{2.5}$$

Now we move to our definition of Ball_t, shown as

$$\text{Ball}_t = \{w : \|w - \hat{w}_t\|_{\Sigma_t}^2 \leq \beta_t\}, \tag{2.6}$$

where β_t is a pre-defined monotonically increasing sequence that we will specify later. Following the "optimism in the face of uncertainty" idea, our ball is centered at $\hat{w}_t$ with β_t being the radius and Σ_t measuring the shape. β_t ensures that the true parameter w^* is always contained in Ball_t w.h.p. In Section 2.5.2, we will show that it suffices to choose

$$\beta_t = \tilde{O}\left(d_w\sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3 t}{\mu^2 T}\right), \tag{2.7}$$

where $\tilde{O}$ hides logarithmic terms in t, T and $1/\delta$ (w.p. $1-\delta$).

Then in Step 5 of Phase II, x_t is selected by joint optimization over $x \in \mathcal{X}$ and $w \in \text{Ball}_t$. Finally, we collect all observations in T rounds and output $\hat{x}$ by uniformly

sampling over $\{x_1, ..., x_T\}$.

2.4.2 Regret Upper Bound

Now we present the cumulative regret upper bound of GO-UCB algorithm.

Theorem 2.4.1 (Cumulative regret of GO-UCB) *Suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold with parameters $F, C_g, C_h, \zeta, \mu, \gamma, \tau, c$. Assume*

$$T > C d_w^2 F^4 \iota^2 \cdot \max\left\{ \frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{\zeta}{\mu c^2} \right\}^2, \tag{2.8}$$

where C is a universal constant and ι is a logarithmic term depending on $n, C_h, 2/\delta$ (both of them from Theorem 2.5.2). Then Algorithm 1 with parameters $n = \sqrt{T}$, $\lambda = C_\lambda \sqrt{T}$ (for a C_λ logarithmically dependent to T and polynomial in all other parameters) and $\beta_{1:T}$ as in eq. (2.7) obeys that with probability at least $1 - \delta$,

$$R_{\sqrt{T}+T} = \tilde{O}\left(\sqrt{T} F + \sqrt{T \beta_T d_w + \frac{T \beta_T^2}{\lambda^2}} \right) = \tilde{O}\left(\frac{d_w^2 \sqrt{T}}{\mu} \right).$$

Let us highlight a few interesting aspects of the result.

Remark 2.4.2 *Without Gaussian process assumption, we propose the first algorithm to solve global optimization problem with $\tilde{O}(\sqrt{T})$ cumulative regret, which is* dimension-free *in terms of its input domain $\mathcal{X}$. GO-UCB is a no-regret algorithm since $\lim_{T\to\infty} R_T/T = 0$, and the output $\hat{x}$ satisfies that $f^* - \mathbb{E}[f(\hat{x})] \leq \tilde{O}(1/\sqrt{T})$, which is also knowns as expected simple regret upper bound. The dependence in T is optimal up to logarithmic factors, as it matches the lower bound for linear bandits [54, Theorem 3].*

Remark 2.4.3 (Choice of λ) *One important deviation from the classical linear bandit analysis is that we require a regularization that centers around $\hat{w}_0$ and the regularization*

weight λ to be $C_\lambda\sqrt{T}$, comparing to $\lambda = O(1)$ in the linear case. The choice is to ensure that $\hat{w}_t$ stays within the local neighborhood of $\hat{w}_0$, and to delicately balance different terms that appear in the regret analysis to ensure that the overall regret bound is $\tilde{O}(\sqrt{T})$.

Remark 2.4.4 (Choice of n) *We choose $n = \sqrt{T}$, therefore, it puts sample complexity requirement on T shown in eq. (2.8). The choice of n plays two roles here. First, it guarantees that the regression result $\hat{w}_0$ lies in the neighboring region of w^* of the loss function $L(w)$ with high probability. The neighboring region of w^* has nice properties, e.g., local strong convexity, which allow us to build the upper bound of ℓ_2-distance between $\hat{w}_0$ and w^*. Second, in Phase I, we are doing uniform sampling over the function so the cumulative regret in Phase I is bounded by $2Fn = 2F\sqrt{T}$ which is at the same $\tilde{O}(\sqrt{T})$ rate as that in Phase II.*

2.5 Proof Overview

In this section, we give a proof sketch of all theoretical results. A key insight of our analysis is that there is more mileage that seminal techniques developed by [16] for analyzing linearly parameterized bandits problems in analyzing non-linear bandits, though we need to localize to a nearly optimal region and carefully handle the non-linear components via more aggressive regularization. Other assumptions that give rise to a similarly good initialization may work too and our new proof can be of independent interest in analyzing other extensions of LinUCB, e.g., to contextual bandits, reinforcement learning and other problems.

In detail, first we prove the estimation error bound of $\hat{w}_0$ for Phase I of GO-UCB algorithm, then prove the feasibility of Ball_t. Finally by putting everything together we prove the cumulative regret bound of GO-UCB algorithm. Due to page limit, we list all auxiliary lemmas in Appendix A and show complete proofs in Section 2.8.

2.5.1 Regression Oracle Guarantee

The goal of Phase I of GO-UCB is to sufficiently explore the unknown objective function with n uniform queries and obtain an estimated parameter $\hat{w}_0$. By assuming access to a regression oracle, we prove the convergence bound of $\hat{w}_0$ w.r.t. w^*, i.e., $\|\hat{w}_0 - w^*\|_2^2$. To get started, we need the following regression oracle lemma.

Lemma 2.5.1 (Adapted from [15])) *Suppose Assumption 2.3.1 & 2.3.2 hold. There is an absolute constant C', such that after round n in Phase I of Algorithm 1, with probability $> 1 - \delta/2$, regression oracle estimated $\hat{w}_0$ satisfies*

$$\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\hat{w}_0) - f_x(w^*))^2] \leq \frac{C' d_w F^2 \iota}{n},$$

where ι is the logarithmic term depending on $n, C_h, 2/\delta$.

[15] proves that expected square error of Empirical Risk Minimization (ERM) estimator can be bounded at the rate of $\tilde{O}(1/n)$ with high probability, rather than $\tilde{O}(1/\sqrt{n})$ rate achieved by Chernoff/Hoeffding bounds. It works with realizable and misspecified settings. Proof of Lemma 2.5.1 includes simplifying it with regression oracle, Assumption 2.3.1, and ε-covering number argument over parameter class. Basically Lemma 2.5.1 says that expected square error of $f_x(\hat{w}_0)$ converges to $f_x(w^*)$ at the rate of $\tilde{O}(1/n)$ with high probability. Based on it, we prove the following regression oracle guarantee.

Theorem 2.5.2 (Regression oracle guarantee) *Suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold. There is an absolute constant C such that after round n in Phase I of Algorithm 1 where n satisfies $n \geq C d_w F^2 \iota \cdot \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{\zeta}{\mu c^2}\right\}$, with probability $> 1 - \delta/2$, regression oracle estimated $\hat{w}_0$ satisfies*

$$\|\hat{w}_0 - w^*\|_2^2 \leq \frac{C d_w F^2 \iota}{\mu n},$$

where ι is the logarithmic term depending on $n, C_h, 2/\delta$.

Compared with Lemma 2.5.1, there is an extra sample complexity requirement on n because we need n to be sufficiently large such that the function can be sufficiently explored and more importantly $\hat{w}_0$ falls into the neighboring region (strongly convex region) of w^*. See Figure 2.1 for illustration. It is also the reason why strong convexity parameter μ appears in the denominator of the upper bound.

2.5.2 Feasibility of Ball_t

The following lemma is the key part of algorithm design of GO-UCB. It says that our definition of Ball_t is appropriate, i.e., throughout all rounds in Phase II, w^* is contained in Ball_t with high probability.

Lemma 2.5.3 (Feasibility of Ball_t) *Set $\Sigma_t, \hat{w}_t$ as in eq. (2.3), (2.5). Set β_t as*

$$\beta_t = \tilde{O}\left(d_w\sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3 t}{\mu^2 T}\right). \tag{2.9}$$

Suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold and choose $n = \sqrt{T}, \lambda = C_\lambda\sqrt{T}$. Then $\forall t \in [T]$ in Phase II of Algorithm 1, w.p. $> 1-\delta$,

$$\|\hat{w}_t - w^*\|_{\Sigma_t}^2 \le \beta_t.$$

For reader's easy reference, we write our choice of β_t again in eq. (2.9). Note this lemma requires careful choices of λ and n because β_t appears later in the cumulative regret bound and β_t is required to be at the rate of $\tilde{O}(1)$. The proof has three steps. First we obtain the closed form solution of $\hat{w}_t$ as in eq. (2.5). Next we use induction to prove that $\forall t \in [T], \|\hat{w}_t - w^*\|_2^2 \le \tilde{O}(\tilde{C}/n)$ for some universal constant $\tilde{C}$. Finally we prove $\|\hat{w}_t - w^*\|_{\Sigma_t}^2 \le \beta_t$.

2.5.3 Regret Analysis

To prove cumulative regrets bound of GO-UCB algorithm, we need following two lemmas of instantaneous regrets in Phase II of GO-UCB.

Lemma 2.5.4 (Instantaneous regret bound) *Set $\Sigma_t, \hat{w}_t, \beta_t$ as in eq. (2.3), (2.5), & (2.7) and suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold, then with probability $> 1-\delta$, w^* is contained in* Ball_t. *Define $u_t = \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$, then $\forall t \in [T]$ in Phase II of Algorithm 1,*

$$r_t \leq 2\sqrt{\beta_t} u_t + \frac{2\beta_t C_h}{\lambda}.$$

The first term of the upper bound is pretty standard, seen also in LinUCB [16] and GP-UCB [14]. After we apply first order gradient approximation of the objective function, the second term is the upper bound of the high order residual term, which introduces extra challenge to derive the upper bound.

Technically, proof of Lemma 2.5.4 requires w^* is contained in our parameter uncertainty ball Ball_t with high probability throughout Phase II of GO-UCB, which has been proven in Lemma 2.5.3. Later, the proof utilizes Taylor's theorem and uses the convexity of Ball_t twice. See Section 2.8.4. The next lemma is an extension of Lemma 2.5.4, where the proof uses monotonically increasing property of β_t in t.

Lemma 2.5.5 (Summation of squared instantaneous regret bound) *Set $\Sigma_t, \hat{w}_t, \beta_t$ as in eq. (2.3), (2.5), & (2.7) and suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold, then with probability $> 1-\delta$, w^* is contained in* Ball_t *and $\forall t \in [T]$ in Phase II of Algorithm 1,*

$$\sum_{t=1}^{T} r_t^2 \leq 16\beta_T d_w \log\left(1 + \frac{TC_g^2}{d_w\lambda}\right) + \frac{8\beta_T^2 C_h^2 T}{\lambda^2}.$$

Proof of Theorem 2.4.1 follows by putting everything together via Cauchy-Shwartz inequality $\sum_{t=1}^{T} r_t \leq \sqrt{T \sum_{t=1}^{T} r_t^2}$.

2.6 Experiments

We compare our GO-UCB algorithm with four Bayesian Optimization (BO) algorithms: GP-EI [29], GP-PI [55], GP-UCB [14], and Trust Region BO (TuRBO) [37], where the first three are classical methods and TuRBO is a more advanced algorithm designed for high-dimensional cases.

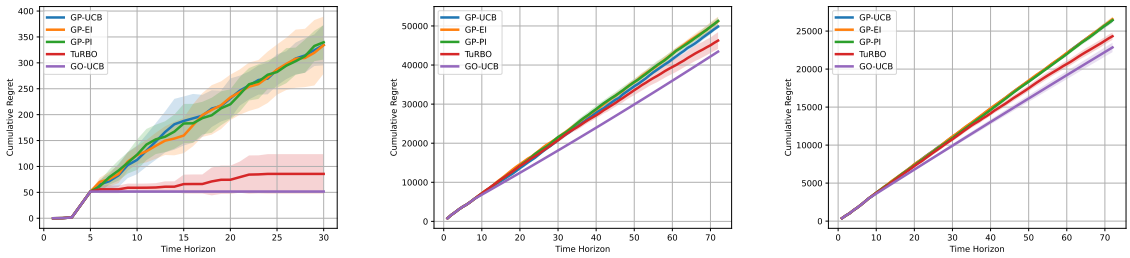
To run GO-UCB, we choose our parametric function model $\hat{f}$ to be a two linear layer neural network with `sigmoid` function being the activation function:

$$\hat{f}(x) = \texttt{linear2}(\texttt{sigmoid}(\texttt{linear1}(x))),$$

where w_1, b_1 denote the weight and bias of `linear1` layer and w_2, b_2 denote those of `linear2` layer. Specifically, we set $w_1 \in \mathbb{R}^{25 \times d_x}, b_1 \in \mathbb{R}^{25}, w_2 \in \mathbb{R}^{25}, b_2 \in \mathbb{R}$, meaning the dimension of activation function is 25. All implementations are based on BoTorch framework [56] and sklearn package [57] with default parameter settings.

2.6.1 Implementation of GO-UCB

Noise parameter $\sigma = 0.01$. Regression oracle in GO-UCB is approximated by stochastic gradient descent algorithm on our two linear layer neural network model with mean squared error loss, 2000 iterations and 10^{-11} learning rate. Exactly solving optimization problem in Step 5 of Phase II may not be computationally tractable, so we use iterative gradient ascent algorithm over x and w with 2000 iterations and 10^{-4} learning rate. β_t is set as $d_w^3 F^4 t/T$. λ is set as $\sqrt{T} \log^2 T$.



(a) f_1 (realizable) (b) f_2 (misspecified) (c) f_3 (misspecified)

Figure 2.2: Cumulative regrets (the lower the better) of all algorithms on 20-dimensional f_1, f_2, f_3 synthetic functions.

2.6.2 Synthetic Experiments

First, we test all algorithms on three high-dimensional synthetic functions defined on $[-5, 5]^{d_x}$ where $d_x = 20$, including both realizable and misspecified cases. The first test function f_1 is created by setting all elements in w_1, b_1, w_2, b_2 in $\hat{f}$ to be 1, so f_1 is a realizable function given $\hat{f}$. The second and third test functions f_2, f_3 are Styblinski-Tang function and Rastrigin function, defined as:

$$f_2 = -\frac{1}{2}\sum_{i=1}^{20} x_i^4 - 16x_i^2 + 5x_i,$$
$$f_3 = -200 + \sum_{i=1}^{20} 10\cos(2\pi x_i) - x_i^2,$$

where x_i denotes the i-th element in its 20 dimensions, so f_2, f_3 are misspecified functions given $\hat{f}$. We set $n = 5, T = 25$ for f_1 and $n = 8, T = 64$ for f_2, f_3. To reduce the effect of randomness in all algorithms, we repeat the whole optimization process for 5 times for all algorithms and report mean and error bar of cumulative regrets. The error bar is measured by Wald's test with 95% confidence, i.e., $1.96\nu/\sqrt{5}$ where ν is standard deviation of cumulative regrets and 5 is the number of repetitions.

From Figure 2.2, we learn that in all tasks our GO-UCB algorithm performs better

than all other four BO approaches. Among BO approches, TuRBO performs the best since it is specifically designed for high-dimensional tasks. In Figure 2.2(a), mean of cumulative regrets of GO-UCB and TuRBO stays the same when $t \geq 22$, which means that both of them have found the global optima, but GO-UCB algorithm is able to find the optimal point shortly after Phase I and enjoys the least error bar. It is well expected since f_1 is a realizable function for $\hat{f}$. Unfortunately, GP-UCB, GP-EI, and GP-PI incur almost linear regrets, showing the bad performances of classical BO algorithms in high-dimensional cases.

In Figure 2.2(b) and 2.2(c), all methods are suffering from linear regrets because f_2, f_3 are misspecified functions. The gap between GO-UCB and other methods is smaller in Figure 2.2(c) than in 2.2(b) because optimizing f_3 is more challenging than f_2 since f_3 has more local optimal points.

2.6.3 Real-World Experiments

To illustrate the GO-UCB algorithm works in real-world tasks, we do hyperparameter tuning experiments on three tasks using three classifiers. Three UCI datasets [58] are Breat-cancer, Australian, and Diabetes, and three classifiers are random forest, multi-layer perceptron, and gradient boosting where each of them has $7, 8, 11$ hyperparameters. For each classifier on each dataset, the function mapping from hyperparameters to classification accuracy is the black-box function that we are maximizing, so the input space dimension $d_x = 7, 8, 11$ for each classifier. We use cumulative regret to evaluate hyperparameter tuning performances, however, best accuracy f^* is unknown ahead of time so we set it to be the best empirical accuracy of each task. To reduce the effect of randomness, we divide each dataset into 5 folds and every time use 4 folds for training and remaining 1 fold for testing. We report mean and error bar of cumulative regrets where error bar

is measured by Wald's test, the same as synthetic experiments.

Hyperparameters can be continuous or categorical, however, in order to fairly compare GO-UCB with Bayesian optimization methods, in all hyperparameter tuning tasks, we set function domain to be $[0, 10]^{d_x}$, a continuous domain. If a hyperparameter is categorical, we allocate equal length domain for each hyperparameter. For example, the seventh hyperparameter of random forest is a bool value, True or False and we define $[0, 5)$ as True and $[5, 10]$ as False. If a hyperparameter is continuous, we set linear mapping from the hyperparameter domain to $[0, 10]$. For example, the sixth hyperparameter of multi-layer perceptron is a float value in $(0, 1)$ thus we multiply it by 10 and map it to $(0, 10)$.

Hyperparameters in hyperparameter tuning tasks. We list hyperparameters in all three tasks as follows.

Classification with Random Forest.

1. Number of trees in the forest, (integer, [20, 200]).
2. Criterion, (string, "gini", "entropy", or "logloss").
3. Maximum depth of the tree, (integer, [1, 10]).
4. Minimum number of samples required to split an internal node, (integer, [2, 10]).
5. Minimum number of samples required to be at a leaf node, (integer, [1, 10]).
6. Maximum number of features to consider when looking for the best split, (string, "sqrt" or "log2").
7. Bootstrap, (bool, True or False).

Classification with Multi-Layer Perceptron.

1. Activation function (string, “identity”, “logistic”, “tanh”, or “relu”).
2. Strength of the L2 regularization term, (float, $[10^{-6}, 10^{-2}]$).
3. Initial learning rate used, (float, $[10^{-6}, 10^{-2}]$).
4. Maximum number of iterations, (integer, [100, 300]).
5. Whether to shuffle samples in each iteration, (bool, True or False).
6. Exponential decay rate for estimates of first moment vector, (float, (0, 1)).
7. Exponential decay rate for estimates of second moment vector (float, (0, 1)).
8. Maximum number of epochs to not meet tolerance improvement, (integer, [1, 10]).

Classification with Gradient Boosting.

1. Loss, (string, “logloss” or “exponential”).
2. Learning rate, (float, (0, 1)).
3. Number of estimators, (integer, [20, 200]).
4. Fraction of samples to be used for fitting the individual base learners, (float, (0, 1)).
5. Function to measure the quality of a split, (string, “friedman mse” or “squared error”).
6. Minimum number of samples required to split an internal node, (integer, [2, 10]).
7. Minimum number of samples required to be at a leaf node, (integer, [1, 10]).
8. Minimum weighted fraction of the sum total of weights, (float, (0, 0.5)).

9. Maximum depth of the individual regression estimators, (integer, [1, 10]).

10. Number of features to consider when looking for the best split, (float, "sqrt" or "log2").

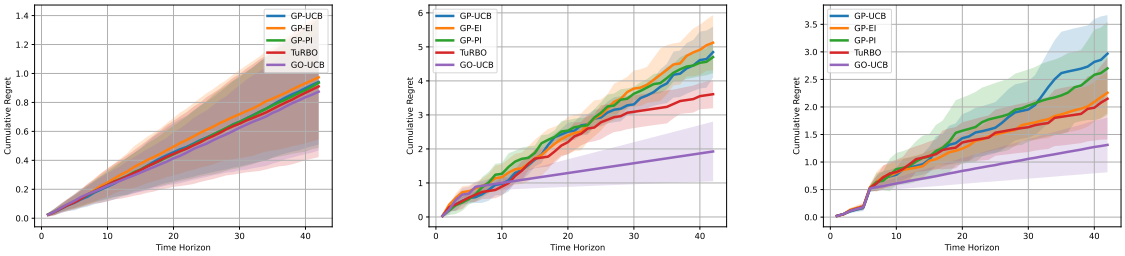
11. Maximum number of leaf nodes in best-first fashion, (integer, [2, 10]).

Figure 2.3 shows results on Breast-cancer dataset. In Figure 2.3(b)(c) GO-UCB performs statistically much better that all other BO algorithms since there is almost no error bar gap between TuRBO and GO-UCB. It shows that GO-UCB can be deployed in real-world applications to replace BO methods. Also, in Figure 2.3(b) performance of GO-UCB Phase I is not good but GO-UCB can still perform better than others in Phase II, which shows the effectiveness of Phase II of GO-UCB. In Figure 2.3(a) all algorithms have similar performances. In Figure 2.3(b), TuRBO performs similarly as GP-UCB, GP-EI, and GP-PI when $t \leq 23$, but after $t = 23$ it performs better and shows a curved regret line by finding optimal points. Results on Australian and Diabetes datasets are shown in Figure 2.4 and 2.5 where similar algorithm performances can be seen.

Note in experiments, we choose parametric model $\hat{f}$ to be a two linear layer neural network. In more real-world experiments, one can choose the model $\hat{f}$ in GO-UCB to be simpler functions or much more complex functions, e.g., deep neural networks, depending on task requirements.

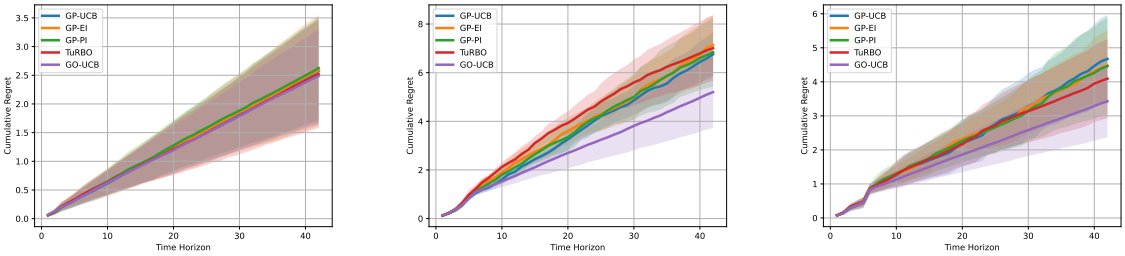
2.7 Conclusions

Global non-convex optimization is an important problem that widely exists in many real-world applications, e.g., deep learning hyper-parameter tuning and new material design. However, solving this optimization problem in general is NP-hard. Existing



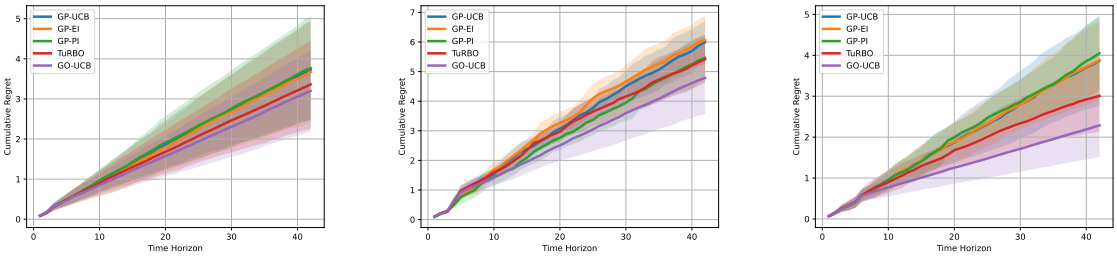
(a) RF ($d_x = 7$) (b) MLP ($d_x = 8$) (c) GB ($d_x = 11$)

Figure 2.3: Cumulative regrets (the lower the better) of all algorithms in real-world hyperparameter tuning task on Breast-cancer dataset.



(a) RF ($d_x = 7$) (b) MLP ($d_x = 8$) (c) GB ($d_x = 11$)

Figure 2.4: Cumulative regrets (the lower the better) of all algorithms in real-world hyperparameter tuning task on Australian dataset.



(a) RF ($d_x = 7$) (b) MLP ($d_x = 8$) (c) GB ($d_x = 11$)

Figure 2.5: Cumulative regrets (the lower the better) of all algorithms in real-world hyperparameter tuning task on Diabetes dataset.

work relies on Gaussian process assumption, e.g., Bayesian optimization, or other non-parametric family which suffers from the curse of dimensionality.

We propose the first algorithm to solve such global optimization with parametric function approximation, which shows a new way of global optimization. GO-UCB first uniformly explores the function and collects a set of observation points and then uses the optimistic exploration to actively select points. At the core of GO-UCB is a carefully designed uncertainty set over parameters based on gradients that allows optimistic exploration. Under realizable parameter class assumption and a few mild geometric conditions, our theoretical analysis shows that cumulative regret of GO-UCB is at the rate of $\tilde{O}(\sqrt{T})$, which is dimension-free in terms of function domain $\mathcal{X}$. Our high-dimensional synthetic test shows that GO-UCB works better than BO methods even in misspecified setting. Moreover, GO-UCB performs better than BO algorithms in real-world hyperparameter tuning tasks, which may be of independent interest.

There is μ, the strongly convexity parameter, in the denominator of upper bound in Theorem 2.4.1. μ can be small in practice, thus the upper bound can be large. Developing the cumulative regret bound containing a term depending on μ but being independent to T remains a future problem.

2.8 Complete Proofs

In this section, we show complete proofs of all technical results in the main sections. For reader's easy reference, we define ι as a logarithmic term depending on $n, C_h, 2/\delta$ (w.p. $> 1 - \delta/2$), ι' as a logarithmic term depending on $t, d_w, C_g, 1/\lambda, 2/\delta$ (w.p. $> 1 - \delta/2$), and ι'' as a logarithmic term depending on $t, d_w, C_g, 1/\lambda$.

2.8.1 Regression Oracle Guarantee

Lemma 2.8.1 (Restatement of Lemma 2.5.1) *Suppose Assumption 2.3.1 & 2.3.2 hold. There is an absolute constant C', such that after round n in Phase I of Algorithm 1, with probability $> 1 - \delta/2$, regression oracle estimated $\hat{w}_0$ satisfies*

$$\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\hat{w}_0) - f_x(w^*))^2] \leq \frac{C' d_w F^2 \iota}{n},$$

where ι is the logarithmic term depending on $n, C_h, 2/\delta$.

Proof: The regression oracle lemma establishes on Lemma A.0.1 which works only for finite function class. In order to work with our continuous parameter class $\mathcal{W}$, we need ε-covering number argument.

First, let $\tilde{w}, \widetilde{\mathcal{W}}$ denote the ERM parameter and finite parameter class after applying covering number argument on $\mathcal{W}$. By Lemma A.0.1, we find that with probability $> 1 - \delta/2$,

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\tilde{w}) - f_x(w^*))^2] \\
&\leq \left(\frac{1+\alpha}{1-\alpha}\right)\left(\inf_{w\in\widetilde{\mathcal{W}}\cup\{w^*\}} \mathbb{E}_{x\sim\mathcal{U}}[(f_x(w) - f_x(w^*))^2] + \frac{F^2\log(|\widetilde{\mathcal{W}}|)\log(2)}{n\alpha}\right) + \frac{2\log(4/\delta)}{n\alpha} \\
&\leq \left(\frac{1+\alpha}{1-\alpha}\right)\left(\frac{F^2\log(|\widetilde{\mathcal{W}}|)\log(2)}{n\alpha}\right) + \frac{2\log(4/\delta)}{n\alpha},
\end{aligned}$$

where the second inequality is by realizable assumption (Assumption 2.3.1). Our parameter class $\mathcal{W} \subseteq [0,1]^{d_w}$, so $\log(|\widetilde{\mathcal{W}}|) = \log(1/\varepsilon^{d_w}) = d_w \log(1/\varepsilon)$ and the new upper bound is that with probability $> 1 - \delta/2$,

$$\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\tilde{w}) - f_x(w^*))^2] \leq C''\left(\frac{d_w F^2 \log(1/\varepsilon)}{n} + \frac{\log(2/\delta)}{n}\right),$$

where C'' is a universal constant obtained by choosing $\alpha = 1/2$. Note $\tilde{w}$ is the ERM parameter in $\widetilde{\mathcal{W}}$ after discretization, not our target parameter $\hat{w}_0 \in \mathcal{W}$. By $(a+b)^2 \leq 2a^2 + 2b^2$,

$$\begin{aligned}\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\hat{w}_0) - f_x(w^*))^2] &\leq 2\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\hat{w}_0) - f_x(\tilde{w}))^2] + 2\mathbb{E}_{x\sim\mathcal{U}}[(f_x(\tilde{w}) - f_x(w^*))^2] \\ &\leq 2\varepsilon^2 C_h^2 + 2C'' \left(\frac{d_w F^2 \log(1/\varepsilon)}{n} + \frac{\log(2/\delta)}{n}\right) \qquad (2.10)\end{aligned}$$

where the second line applies discretization error ε and Assumption 2.3.2. By choosing $\varepsilon = 1/\sqrt{nC_h^2}$, we get

$$(2.10) = \frac{2}{n} + \frac{C'' d_w F^2 \log(nC_h^2)}{n} + \frac{2C'' \log(2/\delta)}{n} \leq C' \frac{d_w F^2 \log(nC_h^2) + \log(2/\delta)}{n}$$

where we can take $C' = 2C''$ (assuming $2 < C'' d_w F^2 \log(nC_h^2)$). The proof completes by defining ι as the logarithmic term depending on $n, C_h, 2/\delta$. ■

Theorem 2.8.2 (Restatement of Theorem 2.5.2) *Suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold. There is an absolute constant C such that after round n in Phase I of Algorithm 1 where n satisfies*

$$n \geq C d_w F^2 \iota \cdot \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{\zeta}{\mu c^2}\right\},$$

with probability $> 1 - \delta/2$, regression oracle estimated $\hat{w}_0$ satisfies

$$\|\hat{w}_0 - w^*\|_2^2 \leq \frac{C d_w F^2 \iota}{\mu n},$$

where ι is the logarithmic term depending on $n, C_h, 2/\delta$.

Proof: Recall the definition of expected loss function $L(w) = \mathbb{E}_{x\sim\mathcal{U}}(f_x(w) - f_x(w^*))^2$

and the second order Taylor's theorem, $L(\hat{w}_0)$ at w^* can be written as

$$L(\hat{w}_0) = L(w^*) + (\hat{w}_0 - w^*)\nabla L(w^*) + \frac{1}{2}\|\hat{w}_0 - w^*\|^2_{\nabla^2 L(\tilde{w})},$$

where $\tilde{w}$ lies between $\hat{w}_0$ and w^*. Also, because $\nabla L(w^*) = \nabla E_{x\sim\mathcal{U}}(f_x(w^*) - f_x(w^*))^2 = 0$, then with probability $> 1 - \delta/2$,

$$\frac{1}{2}\|\hat{w}_0 - w^*\|^2_{\nabla^2 L(\tilde{w})} = L(\hat{w}_0) - L(w^*) \leq \frac{C' d_w F^2 \iota}{n}, \tag{2.11}$$

where the inequality is due to Lemma 2.5.1.

Next, we prove the following lemma stating after a certain number of n samples, $\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)}$ can be bounded by the parameter c from our local-self-concordance assumption.

Lemma 2.8.3 *Suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold. There is an absolute constant C' such that after round n in Phase I of Algorithm 1 where n satisfies*

$$n \geq 2C' d_w F^2 \iota \cdot \max\left\{\frac{\mu^{\gamma/(2-\gamma)}}{\tau^{2/(2-\gamma)}}, \frac{\zeta}{\mu c^2}\right\},$$

then with probability $> 1 - \delta/2$,

$$\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)} \leq c.$$

Proof: First we will prove that when n satisfies the first condition, then $\|\hat{w}_0 - w^*\|_2 \leq (\tau/\mu)^{1/(2-\gamma)}$ by a proof by contradiction.

Assume $\|\hat{w}_0 - w^*\|_2 > (\tau/\mu)^{1/(2-\gamma)}$. Check that under this condition, we have $\frac{\tau}{2}\|\hat{w}_0 - w^*\|_2^\gamma < \frac{\mu}{2}\|\hat{w}_0 - w^*\|_2^2$, therefore the growth-condition (rather than the local strong convexity) part of the Assumption 2.3.3 is active. By the (τ, γ)-growth condition,

we have

$$\frac{\tau}{2}\|\hat{w}_0 - w^*\|_2^\gamma \leq L(\hat{w}_0) - L(w^*) \leq \frac{C' d_w F^2 \iota}{n}.$$

Substituting the first lower bound of n in the assumption, we get

$$\|\hat{w}_0 - w^*\| \leq (\tau/\mu)^{1/(2-\gamma)},$$

thus having a contradiction. This proves that when n satisfies the first condition, $\hat{w}_0$ is within the region where local strong convexity is active.

By the local strong-convexity condition,

$$\frac{\mu}{2}\|\hat{w}_0 - w^*\|_2^2 \leq L(\hat{w}_0) - L(w^*) \leq \frac{C' d_w F^2 \iota}{n}.$$

Then,

$$\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)} \leq \sqrt{\zeta}\|\hat{w}_0 - w^*\|_2 \leq \sqrt{\frac{2\zeta C' d_w F^2 \iota}{\mu n}}.$$

Substitute the second lower bound on n that we assumed, we get that

$$\|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)} \leq \sqrt{\frac{2\zeta C' d_w F^2 \iota}{\mu n}} \leq c.$$

■

Now we continue the proof of Theorem 2.5.2. Observe that $\|\tilde{w} - w^*\|_{\nabla^2 L(w^*)} \leq \|\hat{w}_0 - w^*\|_{\nabla^2 L(w^*)} \leq c$, since $\tilde{w}$ lies on the line-segment between $\hat{w}_0$ and w^*. It follows

that by the c-local self-concordance assumption (Assumption 2.3.3),

$$(1-c)^2\|\hat{w}_0 - w^*\|^2_{\nabla^2 L(w^*)} \leq \|\hat{w}_0 - w^*\|^2_{\nabla^2 L(\tilde{w})}.$$

Therefore, by eq. (2.11)

$$\|\hat{w}_0 - w^*\|^2_{\nabla^2 L(w^*)} \leq \frac{2C' d_w F^2 \iota}{(1-c)^2 n}.$$

The proof completes by inequality $\|\hat{w}_0 - w^*\|_2^2 \leq \|\hat{w}_0 - w^*\|^2_{\nabla^2 L(w^*)}/\mu$ due to μ-strongly convexity of $L(w)$ at w^* (Assumption 2.3.3) and defining $C = 2C'/(1-c)^2$. ■

2.8.2 Properties of Covariance Matrix Σ_t

In eq. (2.3), Σ_t is defined as $\lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i)^\top$. In this section, we prove three lemmas saying the change of Σ_t as $t \in 1, ..., T$ is bounded in Phase II of GO-UCB. The key observation is that at each round i, the change made to Σ_t is $\nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i)^\top$, which is only rank one.

Lemma 2.8.4 (Adapted from [18]) *Set $\Sigma_t, \hat{w}_t$ as in eq. (2.3) & (2.5), suppose Assumption 2.3.1 & 2.3.3 hold, and define $u_t = \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$. Then*

$$\det \Sigma_t = \det \Sigma_0 \prod_{i=0}^{t-1}(1 + u_i^2).$$

Proof: Recall the definition of $\Sigma_t = \lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top$ and we can show that

$$\begin{aligned}
\det \Sigma_{t+1} &= \det(\Sigma_t + \nabla f_{x_t}(w_t) \nabla f_{x_t}(w_t)^\top) \\
&= \det(\Sigma_t^{\frac{1}{2}}(I + \Sigma_t^{-\frac{1}{2}} \nabla f_{x_t}(w_t) \nabla f_{x_t}(w_t)^\top \Sigma_t^{-\frac{1}{2}}) \Sigma_t^{\frac{1}{2}}) \\
&= \det(\Sigma_t) \det(I + \Sigma_t^{-\frac{1}{2}} \nabla f_{x_t}(w_t) (\Sigma_t^{-\frac{1}{2}} \nabla f_{x_t}(w_t))^\top) \\
&= \det(\Sigma_t) \det(I + v_t v_t^\top),
\end{aligned}$$

where $v_t = \Sigma_t^{-\frac{1}{2}} \nabla f_{x_t}(w_t)$. Recall u_t is defined as $\|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$. Because $v_t v_t^\top$ is a rank one matrix, $\det(I + v_t v_t^\top) = 1 + u_t^2$. The proof completes by induction. ■

Lemma 2.8.5 (Adapted from [18]) *Set Σ_t as in eq. (2.3) and suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold. Then*

$$\log\left(\frac{\det \Sigma_{t-1}}{\det \Sigma_0}\right) \leq d_w \log\left(1 + \frac{t C_g^2}{d_w \lambda}\right).$$

Proof of Lemma 2.8.4 directly follows definition of Σ_t and proof of Lemma 2.8.5 involves Lemma 2.8.4 and inequality of arithmetic and geometric means. Note C_g is a constant coming from Assumption 2.3.2. We do not claim any novelty in proofs of these two lemmas which replace feature vector in linear bandit [18] with gradient vectors. *Proof:* Let $\xi_1, ..., \xi_{d_w}$ denote eigenvalues of $\sum_{i=0}^{t-1} \nabla f_{x_i}(w_i) \nabla f_{x_i}(w_i)^\top$, then

$$\sum_{k=1}^{d_w} \xi_k = \operatorname{tr}\left(\sum_{i=0}^{t-1} \nabla f_{x_i}(w_i) \nabla f_{x_i}(w_i)^\top\right) = \sum_{i=0}^{t-1} \|\nabla f_{x_i}(w_i)\|_2^2 \leq t C_g^2, \tag{2.12}$$

where the inequality is by Assumption 2.3.2. By Lemma 2.8.4,

$$\begin{aligned}
\log\left(\frac{\det\Sigma_{t-1}}{\det\Sigma_0}\right) &\leq \log\det\left(I+\frac{1}{\lambda}\sum_{i=0}^{t-1}\nabla f_{x_i}(w_i)\nabla f_{x_i}(w_i)^\top\right)\\
&= \log\left(\prod_{k=1}^{d_w}(1+\xi_k/\lambda)\right)\\
&= d_w\log\left(\prod_{k=1}^{d_w}(1+\xi_k/\lambda)\right)^{1/d_w}\\
&\leq d_w\log\left(\frac{1}{d_w}\sum_{k=1}^{d_w}(1+\xi_k/\lambda)\right)\\
&\leq d_w\log\left(1+\frac{tC_g^2}{d_w\lambda}\right),
\end{aligned}$$

where the second inequality is by inequality of arithmetic and geometric means and the last inequality is due to eq. (2.12). ■

Lemma 2.8.6 *Set $\Sigma_t, \hat{w}_t$ as in eq. (2.3) & (2.5) and suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold. Then*

$$\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)^\top\Sigma_t^{-1}\nabla f_{x_i}(\hat{w}_i) \leq 2d_w\log\left(1+\frac{tC_g^2}{d_w\lambda}\right).$$

A trivial bound of LHS in Lemma 2.8.6 could be simply $O(tC_g^2/\lambda)$. Lemma 2.8.6 is important because it saves the upper bound to be $O(\log(tC_g^2/\lambda))$, which allows us to build a feasible parameter uncertainty ball, shown in the next section.

Proof: First, we prove $\forall i \in \{0,1,...,t-1\}, 0 < \nabla f_{x_i}(\hat{w}_i)^\top\Sigma_t^{-1}\nabla f_{x_i}(\hat{w}_i) < 1$. Recall the definition of Σ_t, it's easy to see that Σ_t is a positive definite matrix and thus $0 <$

$\nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i)$. To prove it's smaller than 1, we need to decompose Σ_t and write

$$\begin{aligned}
&\nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) \\
&= \nabla f_{x_i}(\hat{w}_i)^\top \left(\lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top \right)^{-1} \nabla f_{x_i}(\hat{w}_i) \\
&= \nabla f_{x_i}(\hat{w}_i)^\top \Bigg(\nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top - \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top + \lambda I \\
&\quad + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top \Bigg)^{-1} \nabla f_{x_i}(\hat{w}_i).
\end{aligned}$$

Let $A = -\nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top + \lambda I + \sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top$, and it becomes

$$\nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) = \nabla f_{x_i}(\hat{w}_i)^\top (\nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top + A)^{-1} \nabla f_{x_i}(\hat{w}_i).$$

By applying Sherman-Morrison lemma (Lemma A.0.3), we have

$$\begin{aligned}
&\nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) \\
&= \nabla f_{x_i}(\hat{w}_i)^\top \left(A^{-1} - \frac{A^{-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top A^{-1}}{1 + \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)} \right) \nabla f_{x_i}(\hat{w}_i) \\
&= \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i) - \frac{\nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i) \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)}{1 + \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)} \\
&= \frac{\nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)}{1 + \nabla f_{x_i}(\hat{w}_i)^\top A^{-1} \nabla f_{x_i}(\hat{w}_i)} < 1.
\end{aligned}$$

Next, we use the fact that $\forall x \in (0,1), x \leq 2\log(1+x)$, and we have

$$\begin{aligned}\sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i) &\leq \sum_{i=0}^{t-1} 2\log\left(1 + \nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1} \nabla f_{x_i}(\hat{w}_i)\right) \\ &\leq 2\log\left(\frac{\det \Sigma_{t-1}}{\det \Sigma_0}\right) \\ &\leq 2d_w \log\left(1 + \frac{tC_g^2}{d_w \lambda}\right),\end{aligned}$$

where the last two inequalities are due to Lemma 2.8.4 and 2.8.5. ■

2.8.3 Feasibility of Ball_t

Lemma 2.8.7 (Restatement of Lemma 2.5.3) *Set $\Sigma_t, \hat{w}_t$ as in eq. (2.3), (2.5). Set β_t as*

$$\beta_t = \tilde{O}\left(d_w \sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3 t}{\mu^2 T}\right).$$

Suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold and choose $n = \sqrt{T}, \lambda = C_\lambda \sqrt{T}$. Then $\forall t \in [T]$ in Phase II of Algorithm 1, w.p. $> 1-\delta$,

$$\|\hat{w}_t - w^*\|_{\Sigma_t}^2 \leq \beta_t.$$

Proof: The proof has three steps. First we obtain the closed form solution of $\hat{w}_t$. Next we derive the upper bound of $\|\hat{w}_i - w^*\|_2^2$. Finally we use it to prove that the upper bound of $\|\hat{w}_t - w^*\|_{\Sigma_t}^2$ matches our choice of β_t.

Step 1: Closed form solution of $\hat{w}_t$. The optimal criterion for the objective

function in eq. (2.4) is

$$0 = \lambda(\hat{w}_t - \hat{w}_0) + \sum_{i=0}^{t-1}((\hat{w}_t - \hat{w}_i)^\top \nabla f_{x_i}(\hat{w}_i) + f_{x_i}(\hat{w}_i) - y_i)\nabla f_{x_i}(\hat{w}_i).$$

Rearrange the equation and we have

$$\begin{aligned}
&\lambda(\hat{w}_t - \hat{w}_0) + \sum_{i=0}^{t-1}(\hat{w}_t - \hat{w}_i)^\top \nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i) \\
=& \sum_{i=0}^{t-1}(y_i - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i), \\
&\lambda(\hat{w}_t - \hat{w}_0) + \sum_{i=0}^{t-1}(\hat{w}_t - \hat{w}_i)^\top \nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i) \\
=& \sum_{i=0}^{t-1}(y_i - f_{x_i}(w^*) + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i), \\
&\lambda(\hat{w}_t - \hat{w}_0) + \sum_{i=0}^{t-1}\hat{w}_t^\top \nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i) \\
=& \sum_{i=0}^{t-1}(\hat{w}_i^\top \nabla f_{x_i}(\hat{w}_i) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i), \\
&\hat{w}_t\left(\lambda I + \sum_{i=1}^{t-1}\nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i)^\top\right) - \lambda\hat{w}_0 \\
=& \sum_{i=0}^{t-1}(\hat{w}_i^\top \nabla f_{x_i}(\hat{w}_i) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i), \\
\hat{w}_t\Sigma_t =& \lambda\hat{w}_0 + \sum_{i=0}^{t-1}(\hat{w}_i^\top \nabla f_{x_i}(\hat{w}_i) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i),
\end{aligned}$$

where the second equation is by removing and adding back $f_{x_i}(w^*)$, the third equation is due to definition of observation noise η and the last equation is by our choice of Σ_t (eq.

(2.3)). Now we have the closed form solution of $\hat{w}_t$:

$$\hat{w}_t = \Sigma_t^{-1}\left(\lambda\hat{w}_0 + \sum_{i=0}^{t-1}(\hat{w}_i^\top \nabla f_{x_i}(\hat{w}_i) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\nabla f_{x_i}(\hat{w}_i)\right).$$

Further, $\hat{w}_t - w^*$ can be written as

$$\begin{aligned}
\hat{w}_t - w^* &= \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)(\nabla f_{x_i}(\hat{w}_i)^\top \hat{w}_i + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\right) + \lambda\Sigma_t^{-1}\hat{w}_0 - \Sigma_t^{-1}\Sigma_t w^* \\
&= \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)(\nabla f_{x_i}(\hat{w}_i)^\top \hat{w}_i + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\right) + \lambda\Sigma_t^{-1}(\hat{w}_0 - w^*) \\
&\quad - \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\nabla f_{x_i}(\hat{w}_i)^\top\right) w^* \\
&= \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)(\nabla f_{x_i}(\hat{w}_i)^\top (\hat{w}_i - w^*) + \eta_i + f_{x_i}(w^*) - f_{x_i}(\hat{w}_i))\right) + \lambda\Sigma_t^{-1}(\hat{w}_0 - w^*) \\
&= \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\frac{1}{2}\|w^* - \hat{w}_i\|^2_{\nabla^2 f_{x_i}(\tilde{w})}\right) + \Sigma_t^{-1}\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)\eta_i\right) + \lambda\Sigma_t^{-1}(\hat{w}_0 - w^*),
\end{aligned} \tag{2.13}$$

where the second line is again by our choice of Σ_t and the last equation is by the second order Taylor's theorem of $f_{x_i}(w^*)$ at $\hat{w}_i$ where $\tilde{w}$ lies between w^* and $\hat{w}_i$.

Step 2: Upper bound of $\|\hat{w}_i - w^*\|_2^2$. Note eq. (2.13) holds $\forall i \in [T]$ because all $\hat{w}_i$ are obtained through the same optimization problem, which means

$$\hat{w}_i - w^* = \Sigma_i^{-1}\left(\sum_{\rho=0}^{i-1}\nabla f_{x_\rho}(\hat{w}_\rho)\frac{1}{2}\|w^* - \hat{w}_\rho\|^2_{\nabla^2 f_{x_\rho}(\tilde{w})}\right) + \Sigma_i^{-1}\left(\sum_{\rho=0}^{i-1}\nabla f_{x_\rho}(\hat{w}_\rho)\eta_\rho\right) + \lambda\Sigma_i^{-1}(\hat{w}_0 - w^*).$$

By inequality $(a+b+c)^2 \le 4a^2 + 4b^2 + 4c^2$ and definition of Σ_i, we take the square of

both sides and get

$$\|\hat{w}_i - w^*\|_2^2 \leq \frac{4}{\lambda}\left\|\sum_{\rho=0}^{i-1}\nabla f_{x_\rho}(\hat{w}_\rho)\eta_\rho\right\|_{\Sigma_i^{-1}}^2 + 4\|\hat{w}_0 - w^*\|_2^2 + \frac{1}{\lambda}\left\|\sum_{\rho=0}^{i-1}\nabla f_{x_\rho}(\hat{w}_\rho)\|w^* - \hat{w}_\rho\|_{\nabla^2 f_{x_\rho}(\tilde{w}_\rho)}^2\right\|_{\Sigma_i^{-1}}^2. \tag{2.14}$$

Now we use induction to prove the convergence rate of $\|\hat{w}_i - w^*\|_2^2, \forall i \in [T]$. Recall at the very beginning of Phase II, by Theorem 2.5.2 (check that the condition on n is satisfied due to our condition on T and the choice of $n = \sqrt{T}$), with probability $> 1-\delta/2$,

$$\|\hat{w}_0 - w^*\|_2^2 \leq \frac{Cd_w F^2 \iota}{\mu n}.$$

To derive a claim based on induction, formally, we suppose at round i, there exists some universal constant $\tilde{C}$ such that with probability $> 1 - \delta/2$,

$$\|\hat{w}_i - w^*\|_2^2 \leq \frac{\tilde{C}d_w F^2 \iota}{\mu n}.$$

Our task is to prove that at round $i+1$ with probability $> 1 - \delta/2$,

$$\|\hat{w}_{i+1} - w^*\|_2^2 \leq \frac{\tilde{C}d_w F^2 \iota}{\mu n}.$$

Note $\tilde{C}$ is for induction purpose, which can be different from C.

From eq. (2.14), at round $i+1$ we can write

$$\begin{aligned}
\|\hat{w}_{i+1} - w^*\|_2^2 &\leq \frac{4\sigma^2}{\lambda} \log\left(\frac{\det(\Sigma_i)\det(\Sigma_0)^{-1}}{\delta_i^2}\right) + \frac{4Cd_wF^2\iota}{\mu n} \\
&\quad + \frac{1}{\lambda}\left\|\sum_{\rho=0}^{i} \nabla f_{x_\rho}(\hat{w}_\rho)\|w^* - \hat{w}_\rho\|^2_{\nabla^2 f_{x_\rho}(\tilde{w}_\rho)}\right\|^2_{\Sigma_{i+1}^{-1}} \\
&\leq \frac{4\sigma^2}{\lambda}\left(d_w \log\left(1 + \frac{iC_g^2}{d_w\lambda}\right) + \log\left(\frac{\pi^2 i^2}{3\delta}\right)\right) + \frac{4Cd_wF^2\iota}{\mu n} \\
&\quad + \frac{1}{\lambda}\left\|\sum_{\rho=0}^{i} \nabla f_{x_\rho}(\hat{w}_\rho)\|w^* - \hat{w}_\rho\|^2_{\nabla^2 f_{x_\rho}(\tilde{w}_\rho)}\right\|^2_{\Sigma_{i+1}^{-1}} \\
&\leq \frac{4d_w\sigma^2\iota'}{\lambda} + \frac{4Cd_wF^2\iota}{\mu n} + \frac{1}{\lambda}\left\|\sum_{\rho=0}^{i} \nabla f_{x_\rho}(\hat{w}_\rho)\|w^* - \hat{w}_\rho\|^2_{\nabla^2 f_{x_\rho}(\tilde{w}_\rho)}\right\|^2_{\Sigma_{i+1}^{-1}},
\end{aligned}$$

where the first inequality is due to self-normalized bound for vector-valued martingales (Lemma A.0.2 in Appendix A) and Theorem 2.5.2, the second inequality is by Lemma 2.8.5 and our choice of $\delta_i = 3\delta/(\pi^2 i^2)$, and the last inequality is by defining ι' as the logarithmic term depending on $i, d_w, C_g, 1/\lambda, 2/\delta$ (with probability $> 1 - \delta/2$). The choice of δ_i guarantees the total failure probability over t rounds is no larger than $\delta/2$. Now we use our assumption $\|\hat{w}_i - w^*\|_2^2 \leq \frac{\tilde{C}d_wF^2\iota}{\mu n}$ to bound the last term.

$$\begin{aligned}
\|\hat{w}_{i+1} - w^*\|_2^2 &\leq \frac{4d_w\sigma^2\iota'}{\lambda} + \frac{4Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^2F^4\iota^2}{\mu^2\lambda n^2}\left(\sum_{\rho=0}^{i}\sqrt{\nabla f_{x_\rho}(\hat{w}_\rho)^\top \Sigma_{i+1}^{-1}\nabla f_{x_\rho}(\hat{w}_\rho)}\right)^2 \\
&\leq \frac{4d_w\sigma^2\iota'}{\lambda} + \frac{4Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^2F^4\iota^2}{\mu^2\lambda n^2}\left(\sum_{\rho=0}^{i} 1\right)\left(\sum_{\rho=0}^{i}\nabla f_{x_\rho}(\hat{w}_\rho)^\top \Sigma_{i+1}^{-1}\nabla f_{x_\rho}(\hat{w}_\rho)\right) \\
&\leq \frac{4d_w\sigma^2\iota'}{\lambda} + \frac{4Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^3F^4 i\iota''\iota^2}{\mu^2\lambda n^2},
\end{aligned}$$

where the first inequality is due to smoothness of loss function in Assumption 2.3.3 and triangular inequality, the second inequality is by Cauchy-Schwarz inequality, and the last

inequality is because of Lemma 2.8.6 and defining ι'' as logarithmic term depending on $i, d_w, C_g, 1/\lambda$.

What we need is that there exists some universal constant $\tilde{C}$ such that

$$\frac{4d_w\sigma^2\iota'}{\lambda} + \frac{4Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^3F^4i\iota^2\iota''}{\lambda\mu^2n^2} \leq \frac{\tilde{C}d_wF^2\iota}{\mu n}.$$

Note the LHS is monotonically increasing w.r.t i so the inequality must hold when $i = T$, i.e.,

$$\frac{4d_w\sigma^2\iota'}{\lambda} + \frac{4Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^3F^4T\iota^2\iota''}{\lambda\mu^2n^2} \leq \frac{\tilde{C}d_wF^2\iota}{\mu n}.$$

Recall the range of our function is $[-F, F]$, given any distribution, the variance σ^2 can always be upper bounded by $F^2/4$, so we just need to show that

$$\begin{aligned}
\frac{d_wF^2\iota'}{\lambda} + \frac{4Cd_wF^2\iota}{\mu n} + \frac{\tilde{C}^2C_h^2d_w^3F^4T\iota^2\iota''}{\lambda\mu^2n^2} &\leq \frac{\tilde{C}d_wF^2\iota}{\mu n}, \\
\mu^2n^2\iota' + 4\lambda\mu nC\iota + \tilde{C}^2C_h^2d_w^2F^2T\iota^2\iota'' &\leq \lambda\mu n\tilde{C}\iota, \\
\tilde{C}^2C_h^2d_w^2F^2T\iota^2\iota'' - \tilde{C}\lambda\mu n\iota + \mu^2n^2\iota' + 4\lambda\mu nC\iota &\leq 0,
\end{aligned}$$

where the second and third lines are by rearrangement. A feasible solution on $\tilde{C}$ requires

$$\begin{aligned}
\lambda^2\mu^2n^2\iota^2 - 4C_h^2d_w^2F^2T\iota^2\iota''(\mu^2n^2\iota' + 4\lambda\mu nC\iota) &\geq 0, \\
\lambda^2\mu^2n - 4C_h^2d_w^2F^2T\iota''(\mu^2n\iota' + 4\lambda\mu C\iota) &\geq 0, \qquad (2.15)
\end{aligned}$$

where the second line is by rearrangement. Substitute our choices of $\lambda = C_\lambda\sqrt{T}, n = \sqrt{T}$

and solve the quadratic inequality for C_λ; we get that it suffices to choose

$$C_\lambda = 4\sqrt{C_h^2 d_w^2 F^2 \iota' \iota'' + \frac{16 C^2 C_h^4 d_w^4 F^4 \iota^2 \iota''^2}{\mu^2}} = \tilde{O}\left(\frac{d_w^2}{\mu}\right), \tag{2.16}$$

with assumption $d_w > \mu$. Check that C_λ depends only logarithmically on T and that it ensures eq. (2.15) holds, therefore certifying that a universal constant $\tilde{C}$ exists. Therefore, by induction, we prove that $\forall i \in [T]$ there exists a universal constant $\tilde{C}$ such that with probability $> 1 - \delta/2$,

$$\|\hat{w}_i - w^*\|_2^2 \leq \frac{\tilde{C} d_w F^2 \iota}{\mu n}.$$

With this result, now we are ready to move to **Step 3**.

Step 3: Upper bound of $\|\hat{w}_t - w^*\|_{\Sigma_t}^2$. Multiply both sides of eq. (2.13) by $\Sigma_t^{\frac{1}{2}}$ and we have

$$\Sigma_t^{\frac{1}{2}}(\hat{w}_t - w^*) \leq \frac{1}{2}\Sigma_t^{-\frac{1}{2}}\left(\sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)\|w^* - \hat{w}_i\|_{\nabla^2 f_{x_i}(\tilde{w})}^2\right) + \Sigma_t^{-\frac{1}{2}}\left(\sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)\eta_i\right) + \lambda\Sigma_t^{-\frac{1}{2}}(\hat{w}_0 - w^*).$$

Take square of both sides and by inequality $(a+b+c)^2 \leq 4a^2 + 4b^2 + 4c^2$ we obtain

$$\|\hat{w}_t - w^*\|_{\Sigma_t}^2 \leq 4\left\|\sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)\eta_i\right\|_{\Sigma_t^{-1}}^2 + 4\lambda^2\|\hat{w}_0 - w^*\|_{\Sigma_t^{-1}}^2 + \left\|\sum_{i=0}^{t-1} \nabla f_{x_i}(\hat{w}_i)\|w^* - \hat{w}_i\|_{\nabla^2 f_{x_i}(\tilde{w})}^2\right\|_{\Sigma_t^{-1}}^2.$$

The remaining proof closely follows **Step 2**, i.e.,

$$\begin{aligned}\|\hat{w}_t - w^*\|_{\Sigma_t}^2 &\leq 4d_w\sigma^2\iota' + \frac{4\lambda C d_w F^2\iota}{\mu n} + \frac{\tilde{C}^2 C_h^2 d_w^2 F^4 \iota^2}{\mu^2 n^2}\left(\sum_{i=0}^{t-1}\sqrt{\nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1}\nabla f_{x_i}(\hat{w}_i)}\right)^2 \\ &\leq 4d_w\sigma^2\iota' + \frac{4\lambda C d_w F^2\iota}{\mu n} + \frac{\tilde{C}^2 C_h^2 d_w^2 F^4 \iota^2}{\mu^2 n^2}\left(\sum_{i=0}^{t-1} 1\right)\left(\sum_{i=0}^{t-1}\nabla f_{x_i}(\hat{w}_i)^\top \Sigma_t^{-1}\nabla f_{x_i}(\hat{w}_i)\right) \\ &\leq 4d_w\sigma^2\iota' + \frac{4\lambda C d_w F^2\iota}{\mu n} + \frac{\tilde{C}^2 C_h^2 d_w^3 F^4 t\iota''\iota^2}{\mu^2 n^2} \\ &\leq \tilde{O}\left(d_w\sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3 t}{\mu^2 T}\right),\end{aligned}$$

where the last inequality is by our choices of $\lambda = C_\lambda\sqrt{T}, n = \sqrt{T}$. Therefore, our choice of

$$\beta_t = \tilde{O}\left(d_w\sigma^2 + \frac{d_w^3}{\mu^2} + \frac{d_w^3 t}{\mu^2 T}\right)$$

guarantees that w^* is always contained in Ball_t with probability $1-\delta$. ■

2.8.4 Regret Analysis

Lemma 2.8.8 (Restatement of Lemma 2.5.4) *Set $\Sigma_t, \hat{w}_t, \beta_t$ as in eq. (2.3), (2.5), & (2.7) and suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold, then with probability $> 1-\delta$, w^* is contained in Ball_t. Define $u_t = \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$, then $\forall t \in [T]$ in Phase II of Algorithm 1,*

$$r_t \leq 2\sqrt{\beta_t}u_t + \frac{2\beta_t C_h}{\lambda}.$$

Proof: By definition of instantaneous regret r_t,

$$r_t = f_{x^*}(w^*) - f_{x_t}(w^*).$$

Recall the selection process of x_t and define $\tilde{w} = \text{argmax}_{w \in \text{Ball}_t} f_{x_t}(w)$,

$$r_t \leq f_{x_t}(\tilde{w}) - f_{x_t}(w^*) = (\tilde{w} - w^*)^\top \nabla f_{x_t}(\dot{w}),$$

where the equation is by first order Taylor's theorem and $\dot{w}$ lies between $\tilde{w}$ and w^* which means $\dot{w}$ is guaranteed to be in Ball_t since Ball_t is convex. Then, by adding and removing terms,

$$\begin{aligned}
r_t &= (\tilde{w} - \hat{w}_t + \hat{w}_t - w^*)^\top (\nabla f_{x_t}(\hat{w}_t) - \nabla f_{x_t}(\hat{w}_t) + \nabla f_{x_t}(\dot{w})) \\
&\leq \|\tilde{w} - \hat{w}_t\|_{\Sigma_t} \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}} + \|\hat{w}_t - w^*\|_{\Sigma_t} \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}} + (\tilde{w} - \hat{w}_t)^\top (\nabla f_{x_t}(\dot{w}_t) - \nabla f_{x_t}(\hat{w}_t)) \\
&\quad + (\hat{w}_t - w^*)^\top (\nabla f_{x_t}(\dot{w}) - \nabla f_{x_t}(\hat{w}_t)),
\end{aligned}$$

where the last inequality is due to Holder's inequality. By definitions of β_t in Ball_t and $u_t = \|\nabla f_{x_t}(\hat{w}_t)\|_{\Sigma_t^{-1}}$,

$$r_t \leq 2\sqrt{\beta_t} u_t + (\tilde{w} - \hat{w}_t)^\top (\nabla f_{x_t}(\dot{w}) - \nabla f_{x_t}(\hat{w}_t)) + (\hat{w}_t - w^*)^\top (\nabla f_{x_t}(\dot{w}) - \nabla f_{x_t}(\hat{w}_t)).$$

Again by first order Taylor's theorem where $\ddot{w}$ lies between $\dot{w}$ and $\hat{w}$ and thus $\ddot{w}$ lies in Ball_t,

$$\begin{aligned}
r_t &\leq 2\sqrt{\beta_t} u_t + (\tilde{w} - \hat{w}_t)^\top \Sigma_t^{\frac{1}{2}} \Sigma_t^{-\frac{1}{2}} \nabla^2 f_{x_t}(\ddot{w}) \Sigma_t^{-\frac{1}{2}} \Sigma_t^{\frac{1}{2}} (\dot{w} - \hat{w}_t) \\
&\quad + (\hat{w}_t - w^*)^\top \Sigma_t^{\frac{1}{2}} \Sigma_t^{-\frac{1}{2}} \nabla^2 f_{x_t}(\ddot{w}) \Sigma_t^{-\frac{1}{2}} \Sigma_t^{\frac{1}{2}} (\dot{w} - \hat{w}_t) \\
&\leq 2\sqrt{\beta_t} u_t + \|(\tilde{w} - \hat{w}_t)^\top \Sigma_t^{\frac{1}{2}}\|_2 \|\Sigma_t^{-\frac{1}{2}} \nabla^2 f_{x_t}(\ddot{w}) \Sigma_t^{-\frac{1}{2}}\|_{\text{op}} \|\Sigma_t^{\frac{1}{2}} (\dot{w} - \hat{w}_t)\|_2 \\
&\quad + \|(\hat{w}_t - w^*)^\top \Sigma_t^{\frac{1}{2}}\|_2 \|\Sigma_t^{-\frac{1}{2}} \nabla^2 f_{x_t}(\ddot{w}) \Sigma_t^{-\frac{1}{2}}\|_{\text{op}} \|\Sigma_t^{\frac{1}{2}} (\dot{w} - \hat{w}_t)\|_2 \\
&\leq 2\sqrt{\beta_t} u_t + \frac{2\beta_t C_h}{\lambda},
\end{aligned}$$

where the second inequality is by Holder's inequality and the last inequality is due to definition of β_t in Ball_t, Assumption 2.3.2, and our choice of Σ_t. ■

Lemma 2.8.9 (Restatement of Lemma 2.5.5) *Set $\Sigma_t, \hat{w}_t, \beta_t$ as in eq. (2.3), (2.5), & (2.7) and suppose Assumption 2.3.1, 2.3.2, & 2.3.3 hold, then with probability $> 1-\delta$, w^* is contained in Ball_t and $\forall t \in [T]$ in Phase II of Algorithm 1,*

$$\sum_{t=1}^{T} r_t^2 \leq 16\beta_T d_w \log\left(1 + \frac{TC_g^2}{d_w\lambda}\right) + \frac{8\beta_T^2 C_h^2 T}{\lambda^2}.$$

Proof: By Lemma 2.5.4 and inequality $(a+b)^2 \leq 2a^2 + 2b^2$,

$$\begin{aligned}
\sum_{t=1}^{T} r_t^2 &\leq \sum_{t=1}^{T} 8\beta_t u_t^2 + \frac{8\beta_t^2 C_h^2}{\lambda^2} \\
&\leq 8\beta_T \sum_{i=1}^{T} u_t^2 + \frac{8\beta_T^2 C_h^2 T}{\lambda^2} \\
&\leq 16\beta_T d_w \log\left(1 + \frac{TC_g^2}{d_w\lambda}\right) + \frac{8\beta_T^2 C_h^2 T}{\lambda^2},
\end{aligned}$$

where the second inequality is due to β_t is increasing in t and the last inequality is by Lemma 2.8.6. ■

By putting everything together, we are ready to prove the main cumulative regret theorem.

Proof: [Proof of Theorem 2.4.1] By definition of cumulative regret including both

Phase I and II,

$$\begin{aligned} R_{\sqrt{T}+T} &= \sum_{j=1}^{\sqrt{T}} r_j + \sum_{t=1}^{T} r_t \\ &\leq 2\sqrt{T}F + \sqrt{T\sum_{t=1}^{T} r_t^2} \\ &\leq 2\sqrt{T}F + \sqrt{16T\beta_T d_w \log\left(1 + \frac{TC_g^2}{d_w\lambda}\right) + \frac{8T^2\beta_T^2 C_h^2}{\lambda^2}} \\ &\leq \tilde{O}\left(\sqrt{T}F + \sqrt{T\beta_T d_w + \frac{T^2\beta_T^2}{\lambda^2}}\right), \end{aligned}$$

where the first inequality is due to function range and Cauchy-Schwarz inequality, the second inequality is by Lemma 2.5.5 and the last inequality is obtained by setting $\lambda = C_\lambda\sqrt{T}, n = \sqrt{T}$ as required by Lemma 2.5.3 where C_λ is in eq. (2.16).

Recall that β_t is defined in eq. (2.7), so

$$\beta_T = \tilde{O}\left(\frac{d_w^3}{\mu^2}\right).$$

The proof completes by plugging in upper bound of β_T. ■

Chapter 3

No-Regret Misspecified Linear Bandits

This chapter studies linear bandits when the underlying reward function is *not* linear. Existing work relies on a uniform misspecification parameter ϵ that measures the sup-norm error of the best linear approximation. This results in an unavoidable linear regret whenever $\epsilon > 0$. We describe a more natural model of misspecification which only requires the approximation error at each input x to be proportional to the suboptimality gap at x. It captures the intuition that, for optimization problems, near-optimal regions should matter more and we can tolerate larger approximation errors in suboptimal regions. Quite surprisingly, we show that the classical LinUCB algorithm — designed for the realizable case — is automatically robust against such gap-adjusted misspecification. It achieves a near-optimal $\sqrt{T}$ regret for problems that the best-known regret is almost linear in time horizon T. Technically, our proof relies on a novel self-bounding argument that bounds the part of the regret due to misspecification by the regret itself.

3.1 Introduction

Stochastic linear bandit is a classical problem of online learning and decision-making with many influential applications, e.g., A/B testing [59], recommendation systems [50], advertisement placements [60], clinical trials [61], hyperparameter tuning [62], and new material discovery [63].

More formally, stochastic bandit is a sequential game between an agent who chooses a sequence of actions $x_0, ..., x_{T-1} \in \mathcal{X}$ and nature who decides on a sequence of noisy observations (rewards) $y_0, ..., y_{T-1}$ according to $y_t = f_0(x_t) + \textit{noise}$ for some underlying function f_0. The goal of the learner is to minimize the *cumulative regret* the agent experiences relative to an oracle who knows the best action to choose ahead of time, i.e.,

$$R_T(x_0, ..., x_{T-1}) = \sum_{t=0}^{T-1} r_t = \sum_{t=0}^{T-1} \max_{x \in \mathcal{X}} f_0(x) - f_0(x_t),$$

where r_t is called *instantaneous regret*.

Despite being highly successful in the wild, existing theory for stochastic linear bandits (or more generally learning-oracle based bandits problems [51, 44]) relies on a *realizability* assumption, i.e., the learner is given access to a function class $\mathcal{F}$ such that the true expected reward $f_0 : \mathcal{X} \to \mathbb{R}$ satisfies that $f_0 \in \mathcal{F}$. Realizability is considered one of the strongest and most restrictive assumptions in the standard statistical learning setting, but in the linear bandits, all known attempts to deviate from the realizability assumption result in a regret that grows linearly with T [64, 65, 66, 67, 68, 69].

In practical applications, it is often observed that feature-based representation of the actions with function approximations in estimating the reward can result in very strong policies even if the estimated reward functions are far from being correct [51].

So what went wrong? The critical intuition we rely on is the following:

It should be sufficient for the estimated reward function to clearly *differentiate* good actions from bad ones, rather than requiring it to perfectly estimate the rewards numerically.

Contributions. In this chapter, we formalize this intuition by defining a new family of misspecified bandit problems based on a condition that adjusts the need for an accurate approximation pointwise at every $x \in \mathcal{X}$ according to the suboptimality gap at x. Unlike the existing misspecified linear bandits problems with a linear regret, our problem admits a nearly optimal $\tilde{O}(\sqrt{T})$ regret despite being heavily misspecified. Specifically:

- We define *ρ-gap-adjusted misspecified* (ρ-GAM) function approximations and characterize how they preserve important properties of the true function that are relevant for optimization.

- We show that the classical LinUCB algorithm [16] can be used *as is* (up to some mild hyperparameters) to achieve an $\tilde{O}(\sqrt{T})$ regret under a moderate level of gap-adjusted misspecification ($\rho \leq O(1/\sqrt{\log T})$). In comparison, the regret bound one can obtain under the corresponding uniform misspecification setting is only $\tilde{O}(T/\sqrt{\log T})$. This represents an exponential improvement in the average regret metric R_T/T.

To the best of our knowledge, the suboptimality gap-adjusted misspecification problem was not studied before and we are the first to obtain $\sqrt{T}$-style regrets without a realizability assumption.

Technical novelty. Due to misspecification, we have technical challenges that appear in bounding the instantaneous regret and parameter uncertainty region. We tackle the challenges by a self-bounding trick, i.e., bounding the instantaneous regret by the instantaneous regret itself, which can be of independent interest in more settings, e.g., Gaussian process bandit optimization and reinforcement learning.

3.2 Related Work

The problem of linear bandits was first introduced in [70]. Then [71] proposed the upper confidence bound to study linear bandits where the number of actions is finite. Based on it, [54] proposed an algorithm based on confidence ellipsoids and then [16] simplified the proof with a novel self-normalized martingale bound. Later [50] proposed a simpler and more robust linear bandit algorithm and showed $\tilde{O}(\sqrt{dT})$ regret cannot be improved beyond a polylog factor. [43] further improved the regret upper and lower bound, which characterized the minimax regret up to an iterated logarithmic factor. See [72] for a detailed survey of linear bandits.

In terms of misspecification, [64] first studied the misspecified linear bandit with a fixed action set. They found that LinUCB [16] is not robust when misspecification is large. They showed that in a favourable case when one can test the linearity of the reward function, their RLB algorithm is able to switch between the linear bandit algorithm and finite-armed bandit algorithm to address misspecification issue and achieve the $\tilde{O}(\min\{\sqrt{K}, d\}\sqrt{T})$ regret where K is number of arms.

The most studied setting of model misspecification is uniform misspecification where the ℓ_∞ distance between the best-in-class function and the true function is always upper bounded by some parameter ϵ, i.e.,

Definition 3.2.1 (ϵ-uniform misspecification) *We say function class $\mathcal{F}$ is an ϵ-uniform misspecified approximation of f_0 if there exists $f \in \mathcal{F}$ such that $\sup_{x\in\mathcal{X}} |f(x) - f_0(x)| \leq \epsilon$.*

Under this definition, [65] proposed the optimal design-based phased elimination algorithm for misspecified linear bandits and achieved $\tilde{O}(d\sqrt{T} + \epsilon\sqrt{d}T)$ regret when number of actions is infinite. They also found that with modified confidence band in LinUCB, LinUCB is able to achieve the same regret. With the same misspecification model, [44] studied contextual bandit with regression oracle, [67] studied multi-armed linear contex-

tual bandit, and [66] studied misspecified contextual linear bandits after reduction of the algorithm. All of their results suffer from linear regrets. Later [68] studied misspecified Gaussian process bandit optimization problem and achieved $\tilde{O}(d\sqrt{T}+\epsilon\sqrt{d}T)$ regret when linear kernel is used in Gaussian process. Moreover, their lower bound shows that $\tilde{\Omega}(\epsilon T)$ term is unavoidable in this setting.

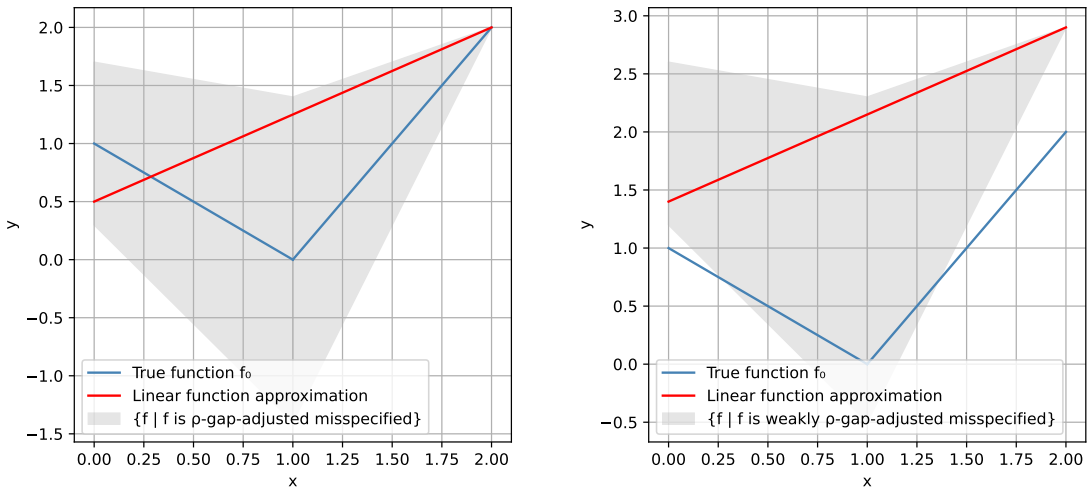
Besides uniform misspecification, there are some work considering different definitions of misspecification. [69] defines misspecification error as an expected squared error between true function and best-in-class function where expectation is taken over distribution of context space and action space. [73] considered average misspecification, which is weaker than uniform misspecification and allows tighter regret bound. However, they also have linear regrets. Our work is different from all related work mentioned above because we are working under a newly defined misspecifiation condition and show that LinUCB is a no-regret algorithm in this case.

Model misspecification is naturally addressed in the related *agnostic* contextual bandits setting [74], but these approaches typically require the action space to be finite, thus not directly applicable to our problem. In addition, empirical evidence [51] suggests that the regression oracle approach works better in practice than the agnostic approach even if realizability cannot be verified.

3.3 Preliminaries

3.3.1 Notations

Let $[n]$ denote the integer set $\{1, 2, ..., n\}$. The algorithm runs in T rounds in total. Let f_0 denote the true function, so the maximum function value is defined as $f^* = \max_{x\in\mathcal{X}} f_0(x)$ and the maximum point is defined as $x^* = \text{argmax}_{x\in\mathcal{X}} f_0(x)$. Let $\mathcal{X} \subset \mathbb{R}^d$



(a) (b)

Figure 3.1: (a): An example of ρ-gap-adjusted misspecification (Definition 3.3.1) in 1-dimension where $\rho = 0.7$. The blue line shows a non-linear true function and the gray region shows the gap-adjusted misspecified function class. Note the vertical range of gray region at a certain point x depends on the suboptimal gap. For example, at $x = 1$ suboptimal gap is 2 and the vertical range is $4\rho = 2.8$. The red line shows a feasible linear function that is able to optimize the true function by taking $x_* = 2$. (b): An example of weak ρ-gap-adjusted misspecification (Definition 3.3.3) in 1-dimension where $\rho = 0.7$. The difference to (a) is that one can shift the qualifying approximation arbitrarily up or down and the specified model only has to ρ-RAM approximate f_0 up to an additive constant factor.

and $\mathcal{Y} \subset \mathbb{R}$ denote the domain and range of f_0. We use $\mathcal{W}$ to denote the parameter class of a family of linear functions $\mathcal{F} := \{f_w : \mathcal{X} \to \mathcal{Y} | w \in \mathcal{W}\}$ where $f_w(x) = w^\top x$. Define w_* as the parameter of best linear approximation function. $\|w\|_2 \leq C_w, \forall w \in \mathcal{W}$ and $\|x\|_2 \leq C_b, \forall x \in \mathcal{X}$. For a vector x, its ℓ_2 norm is denoted by $\|x\|_2 = \sqrt{\sum_{i=1}^d x_i^2}$ and for a matrix A its operator norm is denoted by $\|A\|_{\text{op}}$. For a vector x and a square matrix A, define $\|x\|_A^2 = x^\top A x$.

3.3.2 Problem Setup

We consider the following optimization problem:

$$x_* = \underset{x \in \mathcal{X}}{\operatorname{argmax}} f_0(x),$$

where f_0 is the true function which might not be linear in $\mathcal{X}$. We want to use a linear function $f_w = w^\top x \in \mathcal{F}$ to approximate f_0 and maximize f_0. At time $0 \leq t \leq T-1$, after querying a data point x_t, we will receive a noisy feedback:

$$y_t = f_0(x_t) + \eta_t, \tag{3.1}$$

where η_t is independent, zero-mean, and σ-sub-Gaussian noise.

The major highlight of our study is that we do not rely on the popular *realizability* assumption (*i.e.* $f_0 \in \mathcal{F}$) that is frequently assumed in the existing function approximation literature. Alternatively, we propose the following gap-adjusted misspecification condition.

Definition 3.3.1 (ρ-gap-adjusted misspecification) *We say a function f is a ρ-gap-adjusted misspecified (or ρ-GAM in short) approximation of f_0 if for parameter $0 \leq \rho < 1$,*

$$\sup_{x \in \mathcal{X}} \left| \frac{f(x) - f_0(x)}{f^* - f_0(x)} \right| \leq \rho.$$

We say function class $\mathcal{F} = \{f_w | w \in \mathcal{W}\}$ satisfies ρ-GAM for f_0, if there exists $w^* \in \mathcal{W}$ such that f_{w_*} is a ρ-GAM approximation of f_0.

Observe that when $\rho = 0$, this recovers the standard realizability assumption, but when $\rho > 0$ it could cover many misspecified function classes.

Figure 3.1(a) shows a 1-dimensional example with $f_w(x) = 0.75x + 0.5$ and piece-wise linear function $f_0(x)$ that satisfies local misspecification. With Definition 3.3.1, we have the following proposition.

Proposition 3.3.2 *Let f be a ρ-GAM approximation of f_0 (Definition 3.3.1). Then it holds:*

- *(Preservation of maximizers)*

$$\operatorname*{argmax}_x f(x) = \operatorname*{argmax}_x f_0(x).$$

- *(Preservation of max value)*

$$\max_{x \in \mathcal{X}} f(x) = f^*.$$

- *(Self-bounding property)*

$$|f(x) - f_0(x)| \leq \rho(f^* - f_0(x)) = \rho r(x).$$

This tells f and f_0 coincide on the same global maximum points and the same global maxima if Definition 3.3.1 is satisfied, while allowing f and f_0 to be different (potentially large) at other locations. Therefore, Definition 3.3.1 is a "local" assumption that does not require f to be uniformly close to f_0 (e.g. the "uniform" misspecification assumes $\sup_{x \in \mathcal{X}} |f(x) - f_0(x)| \leq \rho$). Proof of Proposition 3.3.2 is shown in Section 3.6.1.

In addition, we can modify Definition 3.3.1 with a slightly weaker condition that only requires $\operatorname{argmax}_x f(x) = \operatorname{argmax}_x f_0(x)$ but not necessarily $\max_{x \in \mathcal{X}} f(x) = f^*$.

Definition 3.3.3 (Weak ρ-gap-adjusted misspecification) *Denote $f_w^* = \max_{x \in \mathcal{X}} f(x)$. Then we say f is (weak) ρ-gap-adjusted misspecification approximation of f_0 for a pa-*

rameter $0 \leq \rho < 1$ *if:*

$$\sup_{x \in \mathcal{X}} \left| \frac{f(x) - f_w^* + f^* - f_0(x)}{f^* - f_0(x)} \right| \leq \rho.$$

See Figure 3.1(b) for an example satisfying Definition 3.3.3, in which there is a constant gap between f_w^* and f^*. The idea of this weaker assumption is that we can always extend the function class by adding a single offset parameter c w.l.o.g. to learn the constant gap $f^* - f_w^*$. In the linear case, this amounts to homogenizing the feature vector by appending 1. For this reason, we stick to Definition 3.3.1 and linear function approximation for conciseness and clarity in main sections. See Section 3.6.2 for formal statements and proofs of regret bound of linear bandits under Definition 3.3.3.

Note that both Definition 3.3.1 and Definition 3.3.3 are defined generically which do not require any assumptions on the parametric form of f. While we focus on the linear bandit setting in this chapter, this notion can be considered for arbitrary function approximation learning problems.

3.3.3 Assumptions

Assumption 3.3.4 (Boundedness) *For any $x \in \mathcal{X}$, $\|x\|_2 \leq C_b$. For any $w \in \mathcal{W}$, $\|w\|_2 \leq C_w$. Moreover, for any $x, \tilde{x} \in \mathcal{X}$, the true expected reward function $|f_0(x) - f_0(\tilde{x})| \leq F$.*

These are mild assumptions that we assume for convenience. Relaxations of these are possible but not the focus of this chapter. Note that the additional assumption is not required when f_0 is realizable.

Assumption 3.3.5 *Suppose $\mathcal{X} \in \mathbb{R}^d$ is a compact set, and all the global maximizers of*

f_0 live on the $d-1$ dimensional hyperplane. i.e., $\exists a \in \mathbb{R}^d, b \in \mathbb{R}^1$, s.t.

$$\operatorname*{argmax}_{x \in \mathcal{X}} f_0(x) \subset \{x \in \mathbb{R}^d : x^\top a = b\}.$$

For instance, when $d = 1$, the above reduces to that f_0 has a unique maximizer. This is a compatibility assumption for Definition 3.3.1, since any linear function that violates Assumption 3.3.5 will not satisfy Definition 3.3.1.

In addition, to obtain an $\tilde{O}(\sqrt{T})$ regret, for any finite sample T, we require the following condition.

Assumption 3.3.6 (Low misspecification) *The linear function class is a ρ-GAM approximation of f_0 with*

$$\rho < \frac{1}{8d\sqrt{\log\left(1 + \frac{TC_b^2 C_w^2}{d\sigma^2}\right)}} = O\left(\frac{1}{d\sqrt{\log T}}\right). \tag{3.2}$$

The condition is required for technical reasons. Relaxing this condition for LinUCB may require fundamental breakthroughs that knock out logarithmic factors from its regret analysis. This will be further clarified in the proof. In general, however, we conjecture that this condition is not needed and there are algorithms that can achieve $\tilde{O}(\sqrt{T}/(1-\rho))$ regret for any $\rho < 1$, but a new algorithm needs to be designed.

While this assumption may suggest that we still require realizability in a truly asymptotic world, handling a $O(1/\sqrt{\log T})$ level of misspecification is highly non-trivial in finite sample setting. For instance, if T is a trillion, $1/\sqrt{\log(1e12)} \approx 0.19$. This means that for most practical cases, LinUCB is able to tolerate a constant level of misspecification under the GAM model.

3.3.4 LinUCB Algorithm

We will focus on analyzing the classical Linear Upper Confidence Bound (LinUCB) algorithm due to [54, 16], shown below.

Algorithm 2 LinUCB [16]

Input: Predefined sequence β_t for $t = 1, 2, 3, ...$ as in eq. (3.5); Set $\lambda = \sigma^2/C_w^2$ and $\text{Ball}_0 = \mathcal{W}$.

1: **for** $t = 0, 1, 2, ...$ **do**
2: Select $x_t = \text{argmax}_{x \in \mathcal{X}} \max_{w \in \text{Ball}_t} w^\top x$.
3: Observe $y_t = f_0(x_t) + \eta_t$.
4: Update

$$\Sigma_{t+1} = \lambda I + \sum_{i=0}^{t} x_i x_i^\top \text{where } \Sigma_0 = \lambda I. \tag{3.3}$$

5: Update

$$\hat{w}_{t+1} = \underset{w}{\text{argmin}}\, \lambda \|w\|_2^2 + \sum_{i=0}^{t} (w^\top x_i - y_i)_2^2. \tag{3.4}$$

6: Update $\text{Ball}_{t+1} = \{w | \|w - \hat{w}_{t+1}\|_{\Sigma_{t+1}}^2 \leq \beta_{t+1}\}$.
7: **end for**

3.4 Main Results

In this section, we show that the classical LinUCB algorithm [16] works in ρ-gap-adjusted misspecified linear bandits and achieves cumulative regret at the order of $\tilde{O}(\sqrt{T}/(1-\rho))$. The following theorem shows the cumulative regret bound.

Theorem 3.4.1 *Suppose Assumptions 3.3.4, 3.3.5, and 3.3.6 hold. Set*

$$\beta_t = 8\sigma^2 \left(1 + d \log\left(1 + \frac{t C_b^2 C_w^2}{d\sigma^2}\right) + 2 \log\left(\frac{\pi^2 t^2}{3\delta}\right)\right). \tag{3.5}$$

Then Algorithm 2 guarantees w.p. $> 1-\delta$ simultaneously for all $T = 1, 2, ...$

$$R_T \leq F + \sqrt{\frac{8(T-1)\beta_{T-1}d}{(1-\rho)^2}\log\left(1 + \frac{TC_b^2C_w^2}{d\sigma^2}\right)}.$$

Remark 3.4.2 *The result shows that LinUCB achieves $\tilde{O}(\sqrt{T})$ cumulative regret bound and thus it is a no-regret algorithm in ρ-gap-adjusted misspecified linear bandits. In contrast, LinUCB can only achieve $\tilde{O}(\sqrt{T} + \epsilon T)$ regret in uniformly misspecified linear bandits. Even if $\epsilon = \tilde{O}(1/\sqrt{\log T})$, the resulting regret $\tilde{O}(T/\sqrt{\log T})$ is still exponentially worse than ours.*

Proof: By definition of cumulative regret, function range absolute bound F, and Cauchy-Schwarz inequality,

$$\begin{aligned} R_T &= r_0 + \sum_{t=1}^{T-1} r_t \\ &\leq F + \sqrt{\left(\sum_{t=1}^{T-1} 1\right)\left(\sum_{t=1}^{T-1} r_t^2\right)} \\ &= F + \sqrt{(T-1)\sum_{t=1}^{T-1} r_t^2}. \end{aligned}$$

Observe that the choice of β_t is monotonically increasing in t. Also by Lemma 3.4.7, we get that with probability $1-\delta$, $w_* \in \text{Ball}_t, \forall t = 1, 2, 3, ...$, which verifies the condition to apply Lemma 3.4.5 simultaneously for all $T = 1, 2, 3, ...$, thereby completing the proof. ■

3.4.1 Regret Analysis

The proof follows the LinUCB analysis closely. The main innovation is a self-bounding argument that controls the regret due to misspecification by the regret itself. This appears

in Lemma 3.4.4 and then again in the proof of Lemma 3.4.7.

Before we proceed, let Δ_t denote the deviation term of our linear function from the true function at x_t, formally,

$$\Delta_t = f_0(x_t) - w_*^\top x_t, \tag{3.6}$$

And our observation model (eq. (3.1)) becomes

$$y_t = f_0(x_t) + \eta_t = w_*^\top x_t + \Delta_t + \eta_t. \tag{3.7}$$

Moreover, we have the following lemma showing the property of deviation term Δ_t.

Lemma 3.4.3 (Bound of deviation term) $\forall t \in \{0, 1, \ldots, T-1\}$,

$$|\Delta_t| \leq \frac{\rho}{1-\rho} w_*^\top (x_* - x_t).$$

Proof: Recall the definition of deviation term in eq. (3.6):

$$\Delta_t = f_0(x_t) - w_*^\top x_t.$$

By Definition 3.3.1, $\forall t \in \{0, 1, \ldots, T-1\}$,

$$\begin{aligned}
-\rho(f^* - f_0(x_t)) \leq \Delta_t \leq \rho(f^* - f_0(x_t)) \\
-\rho(f^* - w_*^\top x_t - \Delta_t) \leq \Delta_t \leq \rho(f^* - w_*^\top x_t - \Delta_t) \\
-\rho(w_*^\top x_* - w_*^\top x_t - \Delta_t) \leq \Delta_t \leq \rho(w_*^\top x_* - w_*^\top x_t - \Delta_t) \\
\frac{-\rho}{1-\rho}(w_*^\top x_* - w_*^\top x_t) \leq \Delta_t \leq \frac{\rho}{1+\rho}(w_*^\top x_* - w_*^\top x_t),
\end{aligned}$$

where the third line is by Proposition 3.3.2 and the proof completes by taking the absolute

value of the lower and upper bounds. ■

Next, we prove instantaneous regret bound and its sum of squared regret version in the following two lemmas:

Lemma 3.4.4 (Instantaneous regret bound) *Define* $u_t := \|x_t\|_{\Sigma_t^{-1}}$, *assume* $w_* \in \text{Ball}_t$ *then for each* $t \geq 1$

$$r_t \leq \frac{2\sqrt{\beta_t}u_t}{1-\rho}.$$

Proof: By definition of instantaneous regret,

$$\begin{aligned} r_t &= f^* - f_0(x_t) \\ &= w_*^\top x_* - (w_*^\top x_t + \Delta(x_t)) \\ &\leq w_*^\top x_* - w_*^\top x_t + \rho(f^* - f_0(x_t)) \\ &= w_*^\top x_* - w_*^\top x_t + \rho r_t, \end{aligned}$$

where the inequality is by Definition 3.3.1. Therefore, by rearranging the inequality we have

$$r_t \leq \frac{1}{1-\rho}(w_*^\top x_* - w_*^\top x_t) \leq \frac{2\sqrt{\beta_t}u_t}{1-\rho},$$

where the last inequality is by Lemma 3.4.6. ■

Lemma 3.4.5 *Assume* β_t *is monotonically nondecreasing and* $w_* \in \text{Ball}_t$ *for all* $t = 1, ..., T-1$, *then*

$$\sum_{t=1}^{T-1} r_t^2 \leq \frac{8\beta_{T-1}d}{(1-\rho)^2} \log\left(1 + \frac{TC_b^2}{d\lambda}\right).$$

Proof: By definition $u_t = \sqrt{x_t^\top \Sigma_t^{-1} x_t}$ and Lemma 3.4.4,

$$\begin{aligned}
\sum_{t=1}^{T-1} r_t^2 &\leq \sum_{t=1}^{T-1} \frac{4}{(1-\rho)^2} \beta_t u_t^2 \\
&\leq \frac{4\beta_{T-1}}{(1-\rho)^2} \sum_{t=1}^{T-1} u_t^2 \\
&\leq \frac{4\beta_{T-1}}{(1-\rho)^2} \sum_{t=0}^{T-1} u_t^2 \\
&\leq \frac{8\beta_{T-1} d}{(1-\rho)^2} \log\left(1 + \frac{TC_b^2}{d\lambda}\right),
\end{aligned}$$

where the second inequality is by the monotonic increasing property of β_t and the last inequality uses the elliptical potential lemma (Lemma 3.4.9). ■

Previous two lemmas hold on the following lemma, bounding the gap between f^* and the linear function value at x_t, shown below.

Lemma 3.4.6 *Define $u_t = \|x_t\|_{\Sigma_t^{-1}}$ and assume β_t is chosen such that $w_* \in \text{Ball}_t$. Then*

$$w_*^\top (x_* - x_t) \leq 2\sqrt{\beta_t} u_t.$$

Proof: Let $\tilde{w}$ denote the parameter that achieves $\text{argmax}_{w \in \text{Ball}_t} w^\top x_t$, by the optimality of x_t,

$$\begin{aligned}
w_*^\top x_* - w_*^\top x_t &\leq \tilde{w}^\top x_t - w_*^\top x_t \\
&= (\tilde{w} - \hat{w}_t + \hat{w}_t - w_*)^\top x_t \\
&\leq \|\tilde{w} - \hat{w}_t\|_{\Sigma_t} \|x_t\|_{\Sigma_t^{-1}} + \|\hat{w}_t - w_*\|_{\Sigma_t} \|x_t\|_{\Sigma_t^{-1}} \\
&\leq 2\sqrt{\beta_t} u_t
\end{aligned}$$

where the second inequality applies Holder's inequality; the last line uses the definition

of Ball_t (note that both $w_*, \tilde{w} \in \text{Ball}_t$). ■

3.4.2 Confidence Analysis

All analysis in the previous section requires $w_* \in \text{Ball}_t, \forall t \in [T]$. In this section, we show that our choice of β_t in (3.5) is valid and w_* is trapped in the uncertainty set Ball_t with high probability.

Lemma 3.4.7 (Feasibility of Ball_t) *Suppose Assumptions 3.3.4, 3.3.5, and 3.3.6 hold. Set β_t as in eq. (3.5). Then, w.p. $> 1 - \delta$,*

$$\|w_* - \hat{w}_t\|_{\Sigma_t}^2 \leq \beta_t, \forall t = 1, 2, ...$$

Proof: By setting the gradient of objective function in eq. (3.4) to be 0, we obtain the closed form solution of eq. (3.4):

$$\hat{w}_t = \Sigma_t^{-1} \sum_{i=0}^{t-1} y_i x_i.$$

Therefore,

$$\begin{aligned}
\hat{w}_t - w_* &= -w_* + \Sigma_t^{-1} \sum_{i=0}^{t-1} x_i y_i \\
&= -w_* + \Sigma_t^{-1} \sum_{i=0}^{t-1} x_i (x_i^\top w_* + \eta_i + \Delta_i) \\
&= -w_* + \Sigma_t^{-1} \left(\sum_{i=0}^{t-1} x_i x_i^\top \right) w_* + \Sigma_t^{-1} \sum_{i=0}^{t-1} \eta_i x_i + \Sigma_t^{-1} \sum_{i=0}^{t-1} \Delta_i x_i, \qquad (3.8)
\end{aligned}$$

where the second equation is by eq. (3.7) and the first two terms of eq. (3.8) can be further simplified as

$$\begin{aligned}-w_* + \Sigma_t^{-1}\left(\sum_{i=0}^{t-1} x_i x_i^\top\right) w_* &= -w_* + \Sigma_t^{-1}\left(\lambda I + \sum_{i=0}^{t-1} x_i x_i^\top - \lambda I\right) w_* \\ &= -w_* + \Sigma_t^{-1}\Sigma_t w_* - \lambda \Sigma_t^{-1} w_* \\ &= -\lambda \Sigma_t^{-1} w_*,\end{aligned}$$

where the second equation is by definition of Σ_t (eq. (3.3)). Therefore, eq. (3.8) can be rewritten as

$$\hat{w}_t - w_* = -\lambda \Sigma_t^{-1} w_* + \Sigma_t^{-1} \sum_{i=0}^{t-1} \eta_i x_i + \Sigma_t^{-1} \sum_{i=0}^{t-1} \Delta_i x_i.$$

Multiply both sides by $\Sigma_t^{\frac{1}{2}}$ and we have

$$\Sigma_t^{\frac{1}{2}}(\hat{w}_t - w_*) = -\lambda \Sigma_t^{-\frac{1}{2}} w_* + \Sigma_t^{-\frac{1}{2}} \sum_{i=0}^{t-1} \eta_i x_i + \Sigma_t^{-\frac{1}{2}} \sum_{i=0}^{t-1} \Delta_i x_i.$$

Take a square of both sides and apply generalized triangle inequality, we have

$$\|\hat{w}_t - w_*\|_{\Sigma_t}^2 \le 4\lambda^2 \|w_*\|_{\Sigma_t^{-1}}^2 + 4\left\|\sum_{i=0}^{t-1} \eta_i x_i\right\|_{\Sigma_t^{-1}}^2 + 4\left\|\sum_{i=0}^{t-1} \Delta_i x_i\right\|_{\Sigma_t^{-1}}^2. \tag{3.9}$$

The remaining task is to bound these three terms separately. The first term of eq. (3.9) is bounded as

$$4\lambda^2 \|w_*\|_{\Sigma_t^{-1}}^2 \le 4\lambda \|w_*\|_2^2 \le 4\sigma^2,$$

where the first inequality is by definition of Σ_t and $\|\Sigma_t^{-1}\|_{\text{op}} \le 1/\lambda$ and the second

inequality is by choice of $\lambda = \sigma^2/C_w^2$.

The second term of eq. (3.9) can be bounded by Lemma A.0.2 and Lemma A.0.5:

$$4\left\|\sum_{i=0}^{t-1}\eta_i x_i\right\|^2_{\Sigma_t^{-1}} \le 4\sigma^2 \log\left(\frac{\det(\Sigma_t)\det(\Sigma_0)^{-1}}{\delta_t^2}\right) \le 4\sigma^2\left(d\log\left(1+\frac{tC_b^2}{d\lambda}\right) - \log\delta_t^2\right),$$

where δ_t is chosen as $3\delta/(\pi^2 t^2)$ so that the total failure probabilities over T rounds can always be bounded by $\delta/2$:

$$\sum_{t=1}^{T}\frac{3\delta}{\pi^2 t^2} < \sum_{t=1}^{\infty}\frac{3\delta}{\pi^2 t^2} = \frac{3\delta\pi^2}{6\pi^2} = \frac{\delta}{2}.$$

And the third term of eq. (3.9) can be bounded as

$$\begin{aligned}4\left\|\sum_{i=0}^{t-1}\Delta_i x_i\right\|^2_{\Sigma_t^{-1}} &= 4\left(\sum_{i=0}^{t-1}\Delta_i x_i\right)^\top \Sigma_t^{-1}\left(\sum_{j=0}^{t-1}\Delta_j x_j\right)\\ &= 4\sum_{i=0}^{t-1}\sum_{j=0}^{t-1}\Delta_i\Delta_j x_i\Sigma_t^{-1}x_j\\ &\le 4\sum_{i=0}^{t-1}\sum_{j=0}^{t-1}|\Delta_i||\Delta_j|\|x_i\|_{\Sigma_t^{-1}}\|x_j\|_{\Sigma_t^{-1}},\end{aligned}$$

where the last line is by taking the absolute value and Cauchy-Schwarz inequality. Con-

tinue the proof and we have

$$\begin{aligned}
4\sum_{i=0}^{t-1}\sum_{j=0}^{t-1}|\Delta_i||\Delta_j|\|x_i\|_{\Sigma_t^{-1}}\|x_j\|_{\Sigma_t^{-1}} &= 4\left(\sum_{i=0}^{t-1}|\Delta_i|\|x_i\|_{\Sigma_t^{-1}}\right)\left(\sum_{j=0}^{t-1}|\Delta_j|\|x_j\|_{\Sigma_t^{-1}}\right) \\
&= 4\left(\sum_{i=0}^{t-1}|\Delta_i|\|x_i\|_{\Sigma_t^{-1}}\right)^2 \\
&\leq 4\left(\sum_{i=0}^{t-1}|\Delta_i|^2\right)\left(\sum_{i=0}^{t-1}\|x_j\|_{\Sigma_t^{-1}}^2\right) \\
&\leq 4d\rho^2\sum_{i=0}^{t-1}r_i^2.
\end{aligned}$$

where the first inequality is due to Cauchy-Schwarz inequality and the second uses the self-bounding properties $|\Delta_i| \leq \rho r_i$ from Proposition 3.3.2 and Lemma 3.4.8.

To put things together, we have shown that w.p. $> 1-\delta$, for any $t \geq 1$,

$$\|\hat{w}_t - w_*\|_{\Sigma_t^{-1}}^2 \leq 4\sigma^2 + 4\rho^2 d\sum_{i=0}^{t-1}r_i^2 + 4\sigma^2\left(d\log\left(1+\frac{tC_b^2}{d\lambda}\right)+2\log\left(\frac{\pi^2t^2}{3\delta}\right)\right), \quad (3.10)$$

where we condition on (3.10) for the rest of the proof.

Observe that this implies that the feasibility of w_* in Ball_t can be enforced if we choose β_t to be larger than (3.10). The feasiblity of w_* in turn allows us to apply Lemma 3.4.4 to bound the RHS with $\beta_0, ..., \beta_{t-1}$. We will use induction to prove that our choice

$$\beta_t := 2\sigma^2\iota_t \text{ for } t = 1, 2, ...$$

is valid, where short hand

$$\iota_t := 4 + 4\left(d\log\left(1+\frac{tC_b^2}{d\lambda}\right)+2\log\left(\frac{\pi^2t^2}{3\delta}\right)\right).$$

For the base case $t = 1$, by eq. (3.10) and the definition of β_1 we directly have

$\|\hat{w}_1 - w_*\|^2_{\Sigma_1^{-1}} \leq \beta_1$. Assume our choice of β_i is feasible for $i = 1, ..., t-1$, then we can write

$$\begin{aligned}\|\hat{w}_t - w_*\|^2_{\Sigma_t^{-1}} &\leq \sigma^2 \iota_t + 4\rho^2 d \sum_{i=1}^{t-1} \beta_i u_i^2 \\ &\leq \sigma^2 \iota_t + 4\rho^2 d \beta_{t-1} \sum_{i=1}^{t-1} u_i^2,\end{aligned}$$

where the second line is due to non-decreasing property of β_t. Then by Lemma 3.4.9 and Assumption 3.3.6, we have

$$\begin{aligned}\|\hat{w}_t - w_*\|^2_{\Sigma_t^{-1}} &\leq \sigma^2 \iota_t + 8\rho^2 d^2 \beta_{t-1} \log\left(1 + \frac{tC_b^2}{d\lambda}\right) \\ &\leq \sigma^2 \iota_t + \frac{1}{2}\beta_{t-1} \leq 2\sigma^2 \iota_t = \beta_t, \end{aligned} \tag{3.11}$$

The critical difference from the standard LinUCB analysis here is that if β_{t-1} appears on the LHS of the bound and if its coefficient is larger, any valid bound for β_t will have to grow exponentially in t. This is where Assumption 3.3.6 helps us. Assumption 3.3.6 ensures that the coefficient of β_{t-1} is smaller than $1/2$, so we can take $\beta_{t-1} \leq \beta_t$ and move $\beta_t/2$ to the right-hand side. ■

Proof of previous lemma needs the following two lemmas.

Lemma 3.4.8 (Upper bound of $\sum_{i=0}^{t-1} x_i^\top \Sigma_t^{-1} x_i$)

$$\sum_{i=0}^{t-1} x_i^\top \Sigma_t^{-1} x_i \leq d.$$

Proof: Recall that $\Sigma_t = \sum_{i=0}^{t-1} x_i x_i^T + \lambda I_d$.

$$\begin{aligned}\sum_{i=0}^{t-1} x_i^\top \Sigma_t^{-1} x_i &= \sum_{i=0}^{t-1} \text{tr}\left[\Sigma_t^{-1} x_i x_i^T\right] \\ &= \text{tr}\left[\Sigma_t^{-1} \sum_{i=0}^{t-1} x_i x_i^T\right] \\ &= \text{tr}\left[\Sigma_t^{-1}(\Sigma_t - \lambda I_d)\right] \\ &= \text{tr}\left[I_d\right] - \text{tr}\left[\lambda \Sigma_t^{-1}\right] \leq d.\end{aligned}$$

The last line follows from the fact that Σ_t^{-1} is positive semidefinite. ■

Lemma 3.4.9 (Upper bound of $\sum_{i=0}^{t-1} x_i^\top \Sigma_i^{-1} x_i$ (adapted from [16]))

$$\sum_{i=0}^{t-1} x_i^\top \Sigma_i^{-1} x_i \leq 2d \log\left(1 + \frac{tC_b^2}{d\lambda}\right).$$

Proof: First we prove that $\forall i \in \{0, 1, ..., t-1\}, 0 \leq x_i^\top \Sigma_i^{-1} x_i < 1$. Recall the definition of Σ_i and we know Σ_i^{-1} is a positive semidefinite matrix and thus $0 \leq x_i^\top \Sigma_i^{-1} x_i$. To prove $x_i^\top \Sigma_i^{-1} x_i < 1$, we need to decompose Σ_i and write

$$\begin{aligned}x_i^\top \Sigma_i^{-1} x_i &= x_i^\top \left(\lambda I + \sum_{j=0}^{i-1} x_j x_j^\top\right)^{-1} x_i \\ &= x_i^\top \left(x_i x_i^\top - x_i x_i^\top + \lambda I + \sum_{j=0}^{i-1} x_j x_j^\top\right)^{-1} x_i.\end{aligned}$$

Let $A = -x_i x_i^\top + \lambda I + \sum_{j=0}^{i-1} x_j x_j^\top$ and it becomes

$$x_i^\top \Sigma_i^{-1} x_i = x_i^\top (x_i x_i^\top + A)^{-1} x_i.$$

By Sherman-Morrison lemma (Lemma A.0.3), we have

$$\begin{aligned} x_i^\top \Sigma_i^{-1} x_i &= x_i^\top \left(A^{-1} - \frac{A^{-1} x_i x_i^\top A^{-1}}{1 + x_i^\top A^{-1} x_i} \right) x_i \\ &= x_i^\top A^{-1} x_i - \frac{x_i^\top A^{-1} x_i x_i^\top A^{-1} x_i}{1 + x_i^\top A^{-1} x_i} \\ &= \frac{x_i^\top A^{-1} x_i}{1 + x_i^\top A^{-1} x_i} < 1. \end{aligned}$$

Next we use the fact that $\forall x \in [0, 1), x \leq 2\log(x + 1)$ and we have

$$\begin{aligned} \sum_{i=0}^{t-1} x_i^\top \Sigma_i^{-1} x_i &\leq \sum_{i=0}^{t-1} 2\log\left(1 + x_i^\top \Sigma_i^{-1} x_i\right) \\ &\leq 2\log\left(\frac{\det(\Sigma_{t-1})}{\det(\Sigma_0)}\right) \\ &\leq 2d\log\left(1 + \frac{tC_b^2}{d\lambda}\right), \end{aligned}$$

where the last two lines are by Lemma 2.8.4 and Lemma A.0.5. ■

3.5 Conclusions

We study linear bandits with the underlying reward function being non-linear, which falls into the misspecified bandit framework. Existing work on misspecified bandit usually assumes uniform misspecification where the ℓ_∞ distance between the best-in-class function and the true function is upper bounded by the misspecification parameter ϵ. Existing lower bound shows that the $\tilde{\Omega}(\epsilon T)$ term is unavoidable where T is the time horizon, thus the regret bound is always linear. However, in solving optimization problems, one only cares about the approximation error near the global optimal point and approximation error is allowed to be large in highly suboptimal regions. In this chapter, we capture this intuition and define a natural model of misspecification, called ρ-gap-

adjusted misspecificaiton, which only requires the approximation error at each input x to be proportional to the suboptimality gap at x with ρ being the proportion parameter.

Previous work found that classical LinUCB algorithm is not robust in ϵ-uniform misspecified linear bandit when ϵ is large. However, we show that LinUCB is automatically robust against such gap-adjusted misspecification. Under mild conditions, e.g., $\rho \leq O(1/\sqrt{\log T})$, we prove that it achieves the near-optimal $\tilde{O}(\sqrt{T})$ regret for problems that the best-known regret is almost linear. Also, LinUCB doesn't need the knowledge of ρ to run. However, if the upper bound of ρ is revealed to LinUCB, the β_t term can be carefully chosen according to eq. (3.11). Our technical novelty lies in a new self-bounding argument that bounds part of the regret due to misspecification by the regret itself, which can be of independent interest in more settings.

We believe our analysis for LinUCB is tight and the requirement that $\rho = O(1/\sqrt{\log T})$ is essential, but we conjecture that there is a different algorithm that could handle constant ρ or even when ρ approaches 1 at a rate of $O(1/\sqrt{T})$. We leave the resolution to this conjecture as future work.

More broadly, this chapter opens a brand new door for research in model misspecification, including misspecified linear bandits, misspecified kernelized bandits, and even reinforcement learning with misspecified function approximation. Moreover, we hope this chapter make people rethink about the relationship between function optimization and function approximation. In the future, much more can be done. For example, we can design a new no-regret algorithm that works under gap-adjusted misspecification framework where ρ is a constant, and study ρ-gap-adjusted misspecified Gaussian process bandit optimization.

3.6 Additional Proofs

3.6.1 Proof of Proposition 3.3.2

Equivalently, ρ-gap-adjusted misspecification (Definition 3.3.1) satisfies

$$|f(x) - f_0(x)| \leq \rho\,|f^* - f_0(x)|\,, \quad \forall x \in \mathcal{X}. \tag{3.12}$$

Proof: [Proof of preservation of max value: $\max_{x\in\mathcal{X}} f(x) = f^*$]

Let $f_w^* := \max_{x\in\mathcal{X}} f(x)$. We first prove $f_w^* \leq f^*$ by contradiction. Suppose $f_w^* > f^*$, since $\mathcal{X}$ is compact, there exists $x_w \in \mathcal{X}$ such that $f(x_w) = f_w^* > f^*$. Then by eq. (3.12) this implies

$$f(x_w) - f_0(x_w) \leq \rho(f^* - f_0(x_w)) \Rightarrow f^* < f_w^* = f(x_w) \leq \rho f^* + (1-\rho) f_0(x_w) \leq f^*$$

Contraction! Therefore, $f_w^* \leq f^*$. On the other hand, choose $x_0 \in \operatorname{argmax}_{x\in\mathcal{X}} f_0(x)$, then by (3.12) $f(x_0) = f_0(x_0) = f^*$. This implies $f_w^* \geq f^*$. Combing both results to obtain $f_w^* = f^*$. ■

Proof: [Proof of preservation of maximizers: $\operatorname{argmax}_x f(x) = \operatorname{argmax}_x f_0(x)$]

Using that $f(x) \leq \rho f^* + (1-\rho) f_0(x)$ and $\max_{x\in\mathcal{X}} f(x) = f^*$, it is easy to verify $\operatorname{argmax}_x f(x) \subset \operatorname{argmax}_x f_0(x)$. On the other hand, if $x' \in \operatorname{argmax}_x f_0(x)$, then by eq. (3.12) $f(x') = f_0(x') = f^*$ and this means $\operatorname{argmax}_x f_0(x) \subset \operatorname{argmax}_x f(x)$. ■

Proof: [Proof of self-bounding property] This directly comes from the definition. ■

3.6.2 Weak ρ-Gap-Adjusted Misspecification

To study the properties of weak ρ-gap-adjusted misspecification condition, first we recall Definition 3.3.3.

Definition 3.6.1 (Restatement of Weak ρ-gap-adjusted misspecification) *Denote $f_w^* = \max_{x\in\mathcal{X}} f(x)$. Then we say f is (weak) ρ-gap-adjusted misspecification approximation of f_0 for a parameter $0 \le \rho < 1$ if:*

$$\sup_{x\in\mathcal{X}} \left| \frac{f(x) - f_w^* + f^* - f_0(x)}{f^* - f_0(x)} \right| \le \rho.$$

Under the weak ρ-gap-adjusted misspecification condition, it no longer holds $f_w^* = f^*$. However, it still preserves the maximizers.

Proposition 3.6.2 *Under the weak ρ-gap-adjusted misspecification condition, it holds*

$$\operatorname*{argmax}_x f(x) = \operatorname*{argmax}_x f_0(x).$$

Proof: Suppose $x' \in \operatorname{argmax}_x f(x)$, then by definition

$$|f^* - f_0(x')| = |f(x') - f_w^* + f^* - f_0(x')| \le \rho|f^* - f_0(x')| \Rightarrow (1-\rho)|f^* - f_0(x')| \le 0 \Rightarrow x' \in \operatorname*{argmax}_x f_0(x).$$

On the other hand, if $x' \in \operatorname{argmax}_x f_0(x)$, then

$$|f_w^* - f(x')| = |f(x') - f_w^* + f^* - f_0(x')| \le \rho|f^* - f_0(x')| = 0 \Rightarrow x' \in \operatorname*{argmax}_x f(x).$$

■

The next proposition shows the weak ρ-adjusted misspecification condition characterizes the suboptimality gap between f and f_0.

Proposition 3.6.3 *Denote $g(x) := f_w^* - f(x) \ge 0$, $g_0(x) := f^* - f_0(x) \ge 0$, then the weak ρ-gap-adjusted misspecification condition implies:*

$$(1-\rho)g_0(x) \le g(x) \le (1+\rho)g_0(x), \quad x \in \mathcal{X}.$$

This can be proved directly by the triangular inequality. This reveals the weak ρ-gap-adjusted misspecification condition requires $g(x)$ to live in the band $[(1-\rho)g_0(x), (1+\rho)g_0(x)]$, and the concrete maximum values f_w^* and f^* can be arbitrarily different.

To study linear bandits under the weak ρ-gap-adjusted misspecification, we need to slightly modify LinUCB [16] and work with the following LinUCBw algorithm.

Algorithm 3 LinUCBw (adapted from [16])

Input: Predefined sequence β_t for $t = 1, 2, 3, ...$ as in eq. (3.13); Set $\lambda = \sigma^2/C_w^2$ and $\text{Ball}_0 = \mathcal{W}$.

1: **for** $t = 0, 1, 2, ...$ **do**

2: Select $x_t = \text{argmax}_{x\in\mathcal{X}} \max_{[w^\top, c]\in\text{Ball}_t} [w^\top, c] \begin{bmatrix} x \\ 1 \end{bmatrix}$.

3: Observe $y_t = f_0(x_t) + \eta_t$.

4: Update

$$\Sigma_{t+1} = \lambda I_{d+1} + \sum_{i=0}^{t} \begin{bmatrix} x_i \\ 1 \end{bmatrix} \cdot [x_i^\top, 1] \text{ where } \Sigma_0 = \lambda I_{d+1}.$$

5: Update

$$\begin{bmatrix} \hat{w}_{t+1} \\ \hat{c}_{t+1} \end{bmatrix} = \underset{w,c}{\text{argmin}}\, \lambda \left\| \begin{bmatrix} w \\ c \end{bmatrix} \right\|_2^2 + \sum_{i=0}^{t} (w^\top x_i + c - y_i)_2^2.$$

6: Update

$$\text{Ball}_{t+1} = \left\{ \begin{bmatrix} w \\ c \end{bmatrix} \middle| \left\| \begin{bmatrix} w \\ c \end{bmatrix} - \begin{bmatrix} \hat{w}_{t+1} \\ \hat{c}_{t+1} \end{bmatrix} \right\|_{\Sigma_{t+1}}^2 \leq \beta_{t+1} \right\}.$$

7: **end for**

Theorem 3.6.4 *Suppose Assumptions 3.3.4, 3.3.5, and 3.3.6 hold. W.l.o.g., assuming* $c^* = f^* - f_w^* \leq F$. *Set*

$$\beta_t = 8\sigma^2 \left(1 + (d+1) \log \left(1 + \frac{t C_b^2 (C_w^2 + F^2)}{d\sigma^2} \right) + 2 \log \left(\frac{\pi^2 t^2}{3\delta} \right) \right). \tag{3.13}$$

Then Algorithm 3 guarantees w.p. $> 1-\delta$ *simultaneously for all* $T = 1, 2, ...$

$$R_T \leq F + c^* + \sqrt{\frac{8(T-1)\beta_{T-1}(d+1)}{(1-\rho)^2} \log\left(1 + \frac{TC_b^2(C_w^2 + F^2)}{d\sigma^2}\right)}.$$

Remark 3.6.5 *The result again shows that LinUCBw algorithm achieves* $\tilde{O}(\sqrt{T})$ *cumulative regret and thus it is also a no-regret algorithm under the weaker condition (Definition 3.3.3). Note Definition 3.3.3 is quite weak which even doesn't require the true function sits within the approximation function class.*

Proof:

The analysis is similar to the ρ-gap-adjusted case but includes $c^* = f^* - f_w^*$. For instance, let Δ_t^w denote the deviation term of our linear function from the true function at x_t, then

$$\Delta_t^w = f_0(x_t) - w_*^\top x_t - c^*,$$

And our observation model (eq. (3.1)) becomes

$$y_t = f_0(x_t) + \eta_t = w_*^\top x_t + c^* + \Delta_t^w + \eta_t.$$

Then similar to Lemma 3.4.3, we have the following lemma, whose proof is nearly identical to Lemma 3.4.3.

Lemma 3.6.6 (Bound of deviation term) $\forall t \in \{0, 1, \ldots, T-1\}$,

$$|\Delta_t| \leq \frac{\rho}{1-\rho} w_*^\top (x_* - x_t).$$

We also provide the following lemma, which is the counterpart of Lemma 3.4.6.

Lemma 3.6.7 *Define* $u_t = \left\| \begin{bmatrix} x_t \\ 1 \end{bmatrix} \right\|_{\Sigma_t^{-1}}$ *and assume* β_t *is chosen such that* $w_* \in \text{Ball}_t$. *Then*

$$w_*^\top (x_* - x_t) \leq 2\sqrt{\beta_t} u_t.$$

Proof: Let $\tilde{w}, \tilde{c}$ denote the parameter that achieves $\text{argmax}_{w,c \in \text{Ball}_t} w^\top x_t + c$, by the optimality of x_t,

$$\begin{aligned}
w_*^\top x_* - w_*^\top x_t &= \begin{bmatrix} w_*^\top, c^* \end{bmatrix} \begin{bmatrix} x_* \\ 1 \end{bmatrix} - \begin{bmatrix} w_*^\top, c^* \end{bmatrix} \begin{bmatrix} x_t \\ 1 \end{bmatrix} \\
&\leq \begin{bmatrix} \tilde{w}^\top, \tilde{c} \end{bmatrix} \begin{bmatrix} x_t \\ 1 \end{bmatrix} - \begin{bmatrix} w_*^\top, c^* \end{bmatrix} \begin{bmatrix} x_t \\ 1 \end{bmatrix} \\
&= \left(\begin{bmatrix} \tilde{w}^\top, \tilde{c} \end{bmatrix} - \begin{bmatrix} \hat{w}_t^\top, \hat{c}_t \end{bmatrix} + \begin{bmatrix} \hat{w}_t^\top, \hat{c}_t \end{bmatrix} - \begin{bmatrix} w_*^\top, c^* \end{bmatrix} \right) \begin{bmatrix} x_t \\ 1 \end{bmatrix} \\
&\leq \left\| \begin{bmatrix} \tilde{w}^\top, \tilde{c} \end{bmatrix} - \begin{bmatrix} \hat{w}_t^\top, \hat{c}_t \end{bmatrix} \right\|_{\Sigma_t} \left\| \begin{bmatrix} x_t \\ 1 \end{bmatrix} \right\|_{\Sigma_t^{-1}} + \left\| \begin{bmatrix} \hat{w}_t^\top, \hat{c}_t \end{bmatrix} - \begin{bmatrix} w_*^\top, c^* \end{bmatrix} \right\|_{\Sigma_t} \left\| \begin{bmatrix} x_t \\ 1 \end{bmatrix} \right\|_{\Sigma_t^{-1}} \\
&\leq 2\sqrt{\beta_t} u_t
\end{aligned}$$

where the second inequality applies Holder's inequality; the last line uses the definition of Ball_t (note that both $\begin{bmatrix} \tilde{w}^\top, \tilde{c} \end{bmatrix}, \begin{bmatrix} w_*^\top, c^* \end{bmatrix} \in \text{Ball}_t$). ■

The rest of the analysis follows the analysis of Theorem 3.4.1. ■

Chapter 4

Disagreement-Based Active Learning for Privacy Protection

The Private Aggregation of Teacher Ensembles (PATE) framework is one of the most promising recent approaches in differentially private learning. Existing theoretical analysis shows that PATE consistently learns any VC-classes in the realizable setting, but falls short in explaining its success in more general cases where the error rate of the optimal classifier is bounded away from zero. This chapter fills in the gap by introducing the Tsybakov Noise Condition (TNC) and establish stronger and more interpretable learning bounds. These bounds provide new insights into when PATE works and improve over existing results even in the narrower realizable setting. We also investigate the compelling idea of using active learning for saving privacy budget, and empirical studies show the effectiveness of this new idea. The novel components in the proofs include a more refined analysis of the majority voting classifier — which could be of independent interest — and an observation that the synthetic "student" learning problem is nearly realizable by construction under the Tsybakov noise condition.

4.1 Introduction

Differential privacy (DP) [75] is one of the most popular approaches towards addressing the privacy challenges in the era of artificial intelligence and big data. While differential privacy is certainly not a solution to all privacy-related problems, it represents a gold standard and is a key enabler in many applications [76, 77, 78].

Recently, there has been an increasing demand in training machine learning and deep learning models with DP guarantees, which has motivated a growing body of research on this problem [79, 80, 81, 82, 83, 84].

In a nutshell, differentially private machine learning aims at providing formal privacy guarantees that provably reduce the risk of identifying individual data points in the training data, while still allowing the learned model to be deployed and to provide accurate predictions. Many of these methods satisfying DP guarantees work well in low-dimensional regime where the model is small and the data is large. It however remains a fundamental challenge how to avoid the *explicit* dependence in the *ambient dimension* of the model and to develop practical methods in privately releasing deep learning models with a large number of parameters.

The "knowledge transfer" model of differentially private learning is a promising recent development [85, 86] which relaxes the problem by giving the learner access to a public unlabeled dataset. The main workhorse of this model is the Private Aggregation of Teacher Ensembles (PATE) framework:

The *PATE* Framework:

1. Randomly partition the private dataset into K splits.
2. Train one "teacher" classifier on each split.
3. Apply the K "teacher" classifiers on public data and *privately release* their majority votes as pseudo-labels.
4. Output the "student" classifier trained on the pseudo-labeled public data.

PATE achieves DP via the sample-and-aggregate scheme [87] for releasing the pseudo-labels. Since the teachers are trained on disjoint splits of the private dataset, adding or removing one data point could affect only one of the teachers, hence limiting the influence of any single data point. The noise injected in the aggregation will then be able to "obfuscate" the output and obtain provable privacy guarantees.

This approach is appealing in practice as it does not place any restrictions on the *teacher* classifiers, thus allowing any deep learning models to be used in a *model-agnostic* fashion. The competing alternative for differentially private deep learning, NoisySGD [83], is *not* model-agnostic, and it requires significantly more tweaking and modifications to the model to achieve a comparable performance, (e.g., on MNIST), if achievable.

There are a number of DP mechanisms that can be used to instantiate the PATE Framework. Laplace mechanism and Gaussian mechanism are used in [85, 86] respectively. This chapter primarily considers the new mechanism of [88], which instantiates the PATE framework with a more data-adaptive scheme of private aggregation based on the Sparse Vector Technique (SVT). This approach allows PATE to privately label many examples while paying a privacy loss for only a small subset of them (see Algorithm 5 for details). Moreover, [88] provides the first theoretical analysis of PATE which shows that it is able to PAC-learn any hypothesis classes with finite VC-dimension in the realizable

Table 4.1: Summary of our results: excess risk bounds for PATE algorithms.

Algorithm	**PATE** (Gaussian Mech.) [85]	**PATE** (SVT-based) [88]	**PATE** (SVT-based) This chapter	**PATE** (Active Learning) This chapter
Realizable	$\tilde{O}\Big(\frac{d}{(n\epsilon)^{2/3}} \vee \frac{d}{m}\Big)$	$\tilde{O}\Big(\frac{d}{(n\epsilon)^{2/3}} \vee \sqrt{\frac{d}{m}}\Big)$	$\tilde{O}\Big(\frac{d^{3/2}}{n\epsilon} \vee \frac{d}{m}\Big)$	$\tilde{O}\Big(\frac{d^{3/2}\theta^{1/2}}{n\epsilon} \vee \frac{d}{m}\Big)$
τ-TNC	$\tilde{O}\Big(\big(\frac{d^{3/2}}{n\epsilon}\big)^{\frac{2\tau}{4-\tau}} \vee \frac{d}{m}\Big)$	same as agnostic	$\tilde{O}\Big(\big(\frac{d^{3/2}}{n\epsilon}\big)^{\frac{\tau}{2-\tau}} \vee \frac{d}{m}\Big)$	$\tilde{O}\Big(\big(\frac{d^{3/2}\theta^{1/2}}{n\epsilon}\big)^{\frac{\tau}{2-\tau}} \vee \frac{d}{m}\Big)$
Agnostic (vs h^*)	$\Omega(\texttt{Err}(h^*))$ required.	$13\texttt{Err}(h^*)+$ $\tilde{O}\Big(\frac{d^{3/5}}{n^{2/5}\epsilon^{2/5}} \vee \sqrt{\frac{d}{m}}\Big)$	$\Omega(\texttt{Err}(h^*))$ required.	$\Omega(\texttt{Err}(h^*))$ required.
Agnostic (vs $h^{\texttt{agg}}_\infty$)	-	-	Consistent under weaker conditions.	-

- Results new to this chapter are highlighted in blue.
- Teacher number hyperparameter K is chosen optimally. The number of public data points we privately label is chosen optimally (subsampling the available public data to run PATE) to minimize the risk bound. δ is assumed to be in its typical range $\delta < 1/\text{poly}(n)$ and $\epsilon < \log(1/\delta)$. The TNC parameter τ ranges between $(0, 1]$. See Table 4.2 for a checklist of notations.
- Proofs of utility guarantees of PATE (Gaussian mechanism) can be found in Section 4.7.

setting, i.e, expected risk of best hypothesis equals 0. And in this case, the center of teacher agreement is true label. However, this is a giant leap from the standard differentially private learning models (without the access to a public unlabeled dataset) because the VC-classes are *not* privately learnable in general [89, 90]. [88] also establishes a set of results on the agnostic learning setting, albeit less satisfying, as the *excess risk*, i.e., the error rate of the learned classifier relative to the optimal classifier, does not vanish as the number of data points increases, a.k.a., inconsistency.

To fill in the gap, in this chapter, we revisit the problem of model-agnostic private learning in PATE framework in two non-realizable settings: under the Tsybakov Noise Condition (TNC) [91, 92] and in agnostic setting. By making TNC assumption, teachers stay close to the best hypothesis h^* in hypothesis class, thus we consider h^* as the new center for teachers to agree on, instead of considering true label in the realizable setting. We make no assumptions in agnostic setting, and a different center of teacher gravity is considered. In addition, we introduce active learning [93] to PATE and propose a new practical algorithm.

Summary of results. Our contributions are summarized as follows.

1. We show that PATE consistently learns any VC-classes under TNC with fast rates and requires very few unlabeled public data points. When specializing to the realizable case, we show that the sample complexity bound of the SVT-based PATE is $\tilde{O}(d^{3/2}/\alpha\epsilon)$ and $\tilde{O}(d/\alpha)$ for the private and public datasets respectively. The best known results [88] is $\tilde{O}(d^{3/2}/\alpha^{3/2}\epsilon)$ (for private data) and $\tilde{O}(d/\alpha^2)$ (for public data).

2. We analyze standard Gaussian mechanism-based PATE [86] under TNC. In the realizable case, we obtained a sample complexity of $\tilde{O}(d^{3/2}/\alpha\epsilon)$ and $\tilde{O}(d/\alpha)$ for the private and public datasets respectively, which matches the bound of [88] with a simpler and more practical algorithm that uses fewer public data points.

3. We show that PATE learning is *inconsistent* for agnostic learning in general and derive new learning bounds that compete against a sequence of limiting majority voting classifiers.

4. We propose a new active learning-based algorithm, PATE with Active Student Queries (PATE-ASQ), to adaptively select which public data points to release. Under TNC, we show that active learning with standard Gaussian mechanism is able to match the same learning bounds of the SVT-based method for privacy aggregation (Algorithm 4), except some additional dependence.

5. Finally, our experiments on real-life datasets demonstrate that PATE-ASQ achieves significantly better accuracy than standard PATE algorithms while incurring the same or lower privacy loss.

These results (summarized in Table 4.1) provide strong theoretical insight into how PATE works. Interestingly, our theory suggests that *Gaussian mechanism suffices* es-

pecially if we use active learning and that it is better *not* to label all public data when the number of public data points m is large. The remaining data points can be used for semi-supervised learning. These tricks have been proposed in *empirical* studies of PATE (see, e.g., semi-supervised learning [85, 86], active learning [94]), thus our *theory* can be viewed as providing formal justifications to these PATE variants that are producing strong empirical results in *deep learning with differential privacy*.

Motivation and applicability. We conclude the introduction by commenting on the applicability of the knowledge transfer model of differentially private learning and PATE. First, while this model applies only to those cases when a (small) public unlabeled dataset is available, it gains a more favorable privacy-utility tradeoff on those applicable cases. Second, public datasets are often readily available (e.g., census microdata) or can be acquired at a low cost (e.g., incentivizing patients to opt-in) especially if we do not need labels (e.g., getting doctor's diagnosis is expensive). Note that this setting is different from label differential privacy [95] where only labels are considered private. In our problem, even if the public data points are labeled, they are scarce and learning directly from them without using the private data will not give the same learning bound. In addition, PATE uses standard off-the-shelf learners / optimizers as blackboxes, thereby retaining their computational efficiency. For these reasons, we argue that the "knowledge transfer" model is widely applicable and could enable practical algorithms with formal DP guarantees in the many applications where the standard private learning model fails to be sufficiently efficient, private and accurate at the same time.

4.2 Related Work

The literature on differentially private machine learning is enormous and it is impossible for us to provide an exhaustive discussion. Instead we focus on a few closely related work and only briefly discuss other representative results in the broader theory of private learning.

4.2.1 Private Learning with an Auxiliary Public Dataset

The use of an auxiliary unlabeled public dataset was pioneered in empirical studies [85, 86] where PATE was proposed and shown to produce stronger results than NoisySGD in many regimes. Our work builds upon [88]'s first analysis of PATE and substantially improves the theoretical underpinning. To the best of our knowledge, our results are new and we are the first that consider *noise models* and *active learning* for PATE.

[96] also studied the problem of private learning with access to an additional public dataset. Specifically, their result reveals an interesting "theorem of the alternatives"-type result that says either a VC-class is learnable without an auxiliary public dataset, or we need at least $m = \Omega(d/\alpha)$ public data points, which essentially says that our sample complexity on the (unlabeled) public data points are optimal. They also provide an upper bound that says $\tilde{O}(d/\alpha^2)$ private data and $\tilde{O}(d/\alpha)$ public data are sufficient (assuming constant privacy parameter ϵ) to *agnostically learn* any classes with VC-dimension d to α-excess risk. Their algorithm however uses an explicit (distribution-independent) α-net construction due to [97] and exponential mechanism for producing pseudo-labels, which cannot be efficiently implemented. Our contributions are complementary as we focus on *oracle-efficient* algorithms that reduce to the learning bounds of ERM oracles (for passive learning) and active learning oracles. Our algorithms can therefore be implemented (and has been) in practice [85, 86]. Moreover, we show that under TNC, the inefficient

construction is not needed and PATE is indeed consistent and enjoys faster rates. It remains an open problem how to achieve consistent private agnostic learning with only access to ERM oracles.

4.2.2 Privacy-Preserving Prediction

There is another line of work [98] that focuses on the related problem of "privacy-preserving prediction" which does not release the learned model (which we do), but instead privately answer one randomly drawn query x (which we need to answer many, so as to train a model that can be released). While their technique can be used to obtain bounds in our setting, it often involves weaker parameters. More recent works under this model [99, 100] notably achieve consistent agnostic learning in this setting with rates comparable to that of [96]. However, they rely on the same explicit α-net construction [97], which renders their algorithm computationally inefficient in practice. In contrast, we analyze an oracle-efficient algorithm via a reduction to supervised learning (which is practically efficient if we believe supervised learning is easy).

4.2.3 Theory of Private Learning

More broadly, the learnability and sample complexity of private learning were studied under various models in [79, 101, 97, 95, 89, 90, 96]. The VC-classes were shown to be learnable when the either the hypothesis class or the data-domain is finite [79]. [101] characterizes the sample complexity of private learning in the realizable setting with a particular "dimension" that measures the extent to which we can construct a specific discretization of the hypothesis space that works for "all distributions" on data. Such a discretization does not exist, when $\mathcal{H}$ and $\mathcal{X}$ are both continuous. Specifically, the problem of learning threshold functions on $[0, 1]$ having VC-dimension of 1 is not privately

learnable [95, 89].

4.2.4 Weaker Private Learning Models

This setting of private learning was relaxed in various ways to circumvent the above artifact. These include protecting only the labels [95, 97], leveraging prior knowledge with a prior distribution [95], switching to the general learning setting with Lipschitz losses [90], relaxing the distribution-free assumption [90], and the setting we consider in this chapter — when we assume the availability of an auxiliary public data [88, 96]. Note that these settings are closely related to each other in that some additional information about the distribution of the data is needed.

4.2.5 Tsybakov Noise Condition and Statistical Learning Theory

The Tsybakov Noise Condition (TNC) [91, 92] is a natural and well-established condition in learning theory that has long been used in the analysis of passive as well as active learning [102]. The Tsybakov noise condition is known to yield better convergence rates for passive learning [93], and label savings for active learning [103]. However, the contexts under which we use these techniques are different. For instance, while we are making the assumption of TNC, the purpose is not for active learning, but rather to establish stability. When we apply active learning, it is for the synthetic learning problem with pseudo-labels that we release privately, which does not actually satisfy TNC. To the best of our knowledge, we are the first that formally study noise models in the theory of private learning. Lastly, active learning was considered for PATE learning in [94], which demonstrates the clear practical benefits of adaptively selecting what to label. We remain the first that provides theoretical analysis with provable learning bounds.

Table 4.2: Summary of symbols and notations.

Symbol	Definition	Description
$\mathbb{1}(x)$	$= 1(x = \mathtt{T}), = 0(x = \mathtt{F})$	indicator function
$\mathtt{Err}(h)$	$\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbb{1}(h(x) \neq y)]$	expected risk of h w.r.t. $\mathcal{D}$
$\widehat{\mathtt{Err}}(h)$	$\frac{1}{n}\sum_{i=1}^{n}[\mathbb{1}(h(x_i) \neq y_i)]$	empirical risk of h w.r.t. dataset $\{(x_i, y_i)\|i \in [n]\}$
$\mathcal{D}$		distribution over $\mathcal{Z}$
d		VC dimension
$\mathcal{D}_{\mathcal{X}}$		marginal distribution over $\mathcal{X}$
D^T	$\{(x_i^T, y_i^T)\|i \in [n]\} \sim \mathcal{D}$	labeled private teacher dataset
D^S	$\{(x_j^S)\|j \in [m]\} \sim \mathcal{D}_{\mathcal{X}}$	unlabeled public student dataset
$\mathtt{DIS}$		region of disagreement in active learning
$\mathrm{Dis}(h_1, h_2)$	$\mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}[\mathbb{1}(h_1(x) \neq h_2(x)]$	expected disagreement of h_1 and h_2 w.r.t $\mathcal{D}$
$\widehat{\mathrm{Dis}}(h_1, h_2)$	$\frac{1}{n}\sum_{i=1}^{n}[\mathbb{1}(h_1(x_i) \neq h_2(x_i))]$	empirical disagreement of h_1 and h_2 w.r.t. $\{(x_i, y_i)\|i \in [n]\}$
$\mathcal{H}$	$\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$	hypothesis class
h		hypothesis, a function mapping from $\mathcal{X}$ to $\mathcal{Y}$
h^*	$\mathrm{argmin}_{h\in\mathcal{H}}\,\mathtt{Err}(h)$	best hypothesis
$\hat{h}$	$\mathrm{argmin}_{h\in\mathcal{H}}\,\widehat{\mathtt{Err}}(h)$	Empirical Risk Minimizer (ERM)
$\hat{h}^{\mathtt{agg}}$		aggregated classifier in PATE
$\hat{h}^{\mathtt{priv}}$		privately aggregated classifier in PATE
$h_\infty^{\mathtt{agg}}$		infinite ensemble classifier
K		number of teachers
ℓ		labeling budget
m		number of unlabeled student points
n		number of labeled teacher points
$[n]$	$\{1, 2, ..., n\}$	integer set
$\tilde{O}$		big O notation hiding poly-logarithmic factors
$r(x)$	$\mathbb{E}[y\|x]$	regression function from x to y
T		cut-off threshold
$\mathcal{X}$		feature space
$\mathcal{Y}$	$\{0, 1\}$	label space
$\mathcal{Z}$	$\mathcal{X} \times \mathcal{Y}$	sample space
$\mathcal{Z}^*$	$\bigcup_{n\in\mathbb{N}} \mathcal{Z}^n$	space of a dataset of unspecified size
α		excess risk
β, γ		failure probabilities
ϵ, δ	Definition 4.3.1	parameters of differential privacy
ν, ξ	Definition 4.4.12	parameters of high margin condition
τ	Definition 4.4.1	parameter of the Tsybakov noise condition
θ	[93]	disagreement coefficient of active learning
$\widehat{\Delta}$	Eq. 4.1	realized margin
Δ	Eq. 4.4	expected margin
$\perp$		randomly assigned label
$\vee$	$X \vee Y = \max\{X, Y\}$	max operation
$\lesssim, \gtrsim$		inequalities hiding logarithmic factors
c, c', C		constants

4.3 Preliminaries

In this section, first we introduce symbols and notations that we will use throughout this chapter. Then we formally introduce differential privacy and discuss existing progress on PATE and model-agnostic private learning. Finally we introduce disagreement-based active learning, which is the key tool we will use for our new active learning-based PATE algorithm.

4.3.1 Symbols and Notations

We use $[n]$ to denote the set $\{1, 2, ..., n\}$. Let $\mathcal{X}$ denote the feature space, $\mathcal{Y} = \{0, 1\}$ denote the label, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ to denote the sample space, and $\mathcal{Z}^* = \bigcup_{n\in\mathbb{N}} \mathcal{Z}^n$ to denote the space of a dataset of unspecified size. A hypothesis (classifier) h is a function mapping from $\mathcal{X}$ to $\mathcal{Y}$. A set of hypotheses $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ is called the hypothesis class. The VC dimension of $\mathcal{H}$ is denoted by d. Also, let $\mathcal{D}$ denote the distribution over $\mathcal{Z}$, and $\mathcal{D}_{\mathcal{X}}$ denote the marginal distribution over $\mathcal{X}$. $D^T = \{(x_i^T, y_i^T)|i \in [n]\} \sim \mathcal{D}$ is the labeled private teacher dataset, and $D^S = \{(x_j^S)|j \in [m]\} \sim \mathcal{D}_{\mathcal{X}}$ is the unlabeled public student dataset.

The expected risk of a certain hypothesis h with respect to the distribution $\mathcal{D}$ over $\mathcal{Z}$ is defined as $\texttt{Err}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbb{1}(h(x) \neq y)]$, where $\mathbb{1}(x)$ is the indicator function which equals to 1 when x is true, 0 otherwise. The empirical risk of a certain hypothesis h with respect to a dataset $\{(x_i, y_i)|i \in [n]\}$ is defined as $\widehat{\texttt{Err}}(h) = \frac{1}{n}\sum_{i=1}^{n}[\mathbb{1}(h(x_i) \neq y_i)]$. The best hypothesis h^* is defined as $h^* = \text{argmin}_{h\in\mathcal{H}} \texttt{Err}(h)$, and the Empirical Risk Minimizer (ERM) $\hat{h}$ is defined as $\hat{h} = \text{argmin}_{h\in\mathcal{H}} \widehat{\texttt{Err}}(h)$. $\hat{h}^{\texttt{agg}}$ is used to denote the aggregated classifier in the PATE framework. $\hat{h}^{\texttt{priv}}$ denotes the privately aggregated one. The expected disagreement between a pair of hypotheses h_1 and h_2 with respect to the distribution $\mathcal{D}_{\mathcal{X}}$ is defined as $\text{Dis}(h_1, h_2) = \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}[\mathbb{1}(h_1(x) \neq h_2(x)]$. The empirical

disagreement between a pair of hypotheses h_1 and h_2 with respect to a dataset $\{(x_i, y_i)|i \in [n]\}$ is defined as $\widehat{\text{Dis}}(h_1, h_2) = \frac{1}{n}\sum_{i=1}^{n}[\mathbb{1}(h_1(x_i) \neq h_2(x_i))]$. Throughout this chapter, we use standard big O notations; and to improve the readability, we use $\lesssim$ and $\tilde{O}$ to hide poly-logarithmic factors. For reader's easy reference, we summarize the symbol and notations above in Table 4.2.

4.3.2 Differential Privacy and Private Learning

Now we formally introduce differential privacy.

Definition 4.3.1 (Differential Privacy [104]) *A randomized algorithm $\mathcal{M} : \mathcal{Z}^* \to \mathcal{R}$ is (ϵ, δ)-DP (differentially private) if for every pair of* neighboring *datasets $D, D' \in \mathcal{Z}^*$ (denoted by $\|D - D'\|_1 = 1$) for all $\mathcal{S} \subseteq \mathcal{R}$:*

$$\mathbb{P}(\mathcal{M}(D) \in \mathcal{S}) \leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{M}(D') \in \mathcal{S}) + \delta.$$

The definition says that if an algorithm $\mathcal{M}$ is DP, then no adversary can use the output of $\mathcal{M}$ to distinguish between two parallel worlds where an individual is in the dataset or not. ϵ, δ are privacy loss parameters that quantify the strength of the DP guarantee. The closer they are to 0, the stronger the guarantee is.

The problem of DP learning aims at designing a randomized training algorithm that satisfies Definition 4.3.1. More often than not, the research question is about understanding the privacy-utility trade-offs and characterizing the Pareto optimal frontiers.

4.3.3 PATE and Model-Agnostic Private Learning

There are different ways we can instantiate the PATE framework to privately aggregate the teachers' predicted labels. The simplest, described in Algorithm 4, uses Gaussian

Algorithm 4 Standard PATE [86]

Input: "Teachers" $\hat{h}_1, ..., \hat{h}_K$ trained on *disjoint* subsets of the private data. "Nature" chooses an *adaptive* sequence of data points $x_1, ..., x_\ell$. Privacy parameters $\epsilon, \delta > 0$.

1: Find σ such that $\sqrt{\frac{2\ell \log(1/\delta)}{\sigma^2}} + \frac{\ell}{2\sigma^2} = \epsilon$.
2: Nature chooses x_1.
3: **for** $j \in [\ell]$ **do**
4: Output $\hat{y}_j \leftarrow \mathbb{1}(\sum_{k=1}^{K} \hat{h}_k(x_j) + \mathcal{N}(0, \sigma^2) \geq K/2)$.
5: Nature chooses x_{j+1} adaptively (as a function of the output vector till time j).
6: **end for**

mechanism to perturb the voting score.

An alternative approach due to [88] uses the Sparse Vector Technique (SVT) in a nontrivial way to privately label substantially more data points in the cases when teacher ensemble's predictions are *stable* for most input data. The stability is quantified in terms of the margin function, defined as

$$\widehat{\Delta}(x) := \Big| 2 \sum_{k=1}^{K} \hat{h}_k(x) - K \Big|, \tag{4.1}$$

which measures the absolute value of the difference between the number of votes (see Algorithm 5).

In both algorithms, the privacy budget parameters ϵ, δ are taken as an input and the following privacy guarantee applies to all input datasets.

Theorem 4.3.2 *Algorithm 4 and 5 are both (ϵ, δ)-DP.*

Careful readers may note the slightly improved constants in the formula for calibrating privacy than when these methods were first introduced. We include the new proof based on the *concentrated differential privacy* [105] approach in the Section 4.7.

The key difference between the two private-aggregation mechanisms is that the standard PATE pays for a unit privacy loss for every public data point labeled, while the

Algorithm 5 SVT-based PATE [88]

Input: "Teacher" classifiers $\hat{h}_1, ..., \hat{h}_K$ trained on *disjoint* subsets of the private data. "Nature" chooses an *adaptive* sequence of data points $x_1, ..., x_\ell$. Unstable cut-off T, privacy parameters $\epsilon, \delta > 0$.

Nature chooses x_1.
$\lambda \leftarrow (\sqrt{2T(\epsilon + \log(2/\delta))} + \sqrt{2T\log(2/\delta)})/\epsilon$.
$w \leftarrow 3\lambda \log(2(\ell + T)/\delta), \hat{w} \leftarrow w + \texttt{Lap}(\lambda)$.
$c = 0$.
for $j \in [\ell]$ **do**
  $\text{dist}_j \leftarrow \max\{0, \lceil \widehat{\Delta}(x_j)/2 \rceil - 1\}$.
  $\widehat{\text{dist}_j} \leftarrow \text{dist}_j + \texttt{Lap}(2\lambda)$.
  if $\widehat{\text{dist}_j} > \hat{w}$ **then**
    Output $\hat{y}_j \leftarrow \mathbb{1}(\sum_{k=1}^{K} \hat{h}_k(x_j) \geq K/2)$.
  else
    Output $\hat{y}_j \leftarrow \perp$.
    $c \leftarrow c + 1$, break if $c \geq T$.
    $\hat{w} \leftarrow w + \texttt{Lap}(\lambda)$.
  end if
  Nature chooses x_{j+1} adaptively (based on $\hat{y}_1, ..., \hat{y}_j$).
end for

SVT-based PATE essentially pays only for those queries where the voted answer from the teacher ensemble is close to be unstable (those with a small margin). Combining this intuition with the fact that the individual classifiers are accurate — by the statistical learning theory, they are — the corresponding majority voting classifier can be shown to be accurate with a large margin. These two critical observations of [88] lead to the first learning theoretic guarantees for SVT-based PATE. For completeness, we include this result with a concise new proof in Section 4.7.

Lemma 4.3.3 (Adapted from Theorem 3.11 of [106]) *If the classifiers $\hat{h}_1, ..., \hat{h}_K$ and the sequence $x_1, ..., x_\ell$ obey that there are at most T of them such that $\widehat{\Delta}(x_k) < K/3$ for $K = 136\log(4\ell T/\min(\delta, \beta/2)) \cdot \sqrt{T\log(2/\delta)}/\epsilon$. Then with probability at least $1 - \beta$, Algorithm 5 finishes all ℓ queries and for all $i \in [\ell]$ such that $\widehat{\Delta}(x_i) \geq K/3$, the output of Algorithm 5 is $\hat{h}^{\texttt{agg}}(x_i)$.*

Lemma 4.3.4 (Lemma 4.2 of [106]) *If the classifiers $\hat{h}_1, ..., \hat{h}_K$ obey that each of them makes at most B mistakes on data $(x_1, y_1), ..., (x_\ell, y_\ell)$, then*

$$\left| \left\{ i \in [\ell] \;\middle|\; \sum_{k=1}^{K} \mathbb{1}(\hat{h}_k(x_i) \neq y_i) \geq \frac{K}{3} \right\} \right| \leq 3B.$$

Lemma 4.3.4 implies that if the individual classifiers are accurate — by the statistical learning theory, they are — the corresponding majority voting classifier is not only nearly as accurate, but also has sufficiently large margin that satisfies the conditions in Lemma 4.3.3.

Next, we state and provide a straightforward proof of the following results due to [106]. The results are already stated in the referenced work in the form of sample complexities, but we include a more direct analysis of the error bound and clarify a few technical subtleties.

Algorithm 6 PATE-PSQ

Input: Labeled private teacher dataset D^T, unlabeled public student dataset D^S, unstable query cutoff T, privacy parameters $\epsilon, \delta > 0$; number of splits K.

1: Randomly and evenly split the teacher dataset D^T into K parts $D_k^T \subseteq D^T$ where $k \in [K]$.
2: Train K classifiers $\hat{h}_k \in \mathcal{H}$, one from each part D_k^T.
3: Call Algorithm 5 with parameters $(\hat{h}_1, ..., \hat{h}_K), D^S, T, \epsilon, \delta$ and $\ell = m$ to obtain pseudo-labels for the public dataset $\hat{y}_1^S, ..., \hat{y}_m^S$. (Alternatively, call Algorithm 4 with parameters $(\hat{h}_1, ..., \hat{h}_K), D^S, \epsilon, \delta, \ell = m$)
4: For those pseudo labels that are $\perp$, assign them arbitrarily to $\{0, 1\}$.

Output: $\hat{h}^S$ trained on pseudo-labeled student dataset.

Theorem 4.3.5 (Adapted from Theorems 4.6 and 4.7 of [106]) *Set*

$$T = 3\Big(\mathbb{E}[\texttt{Err}(\hat{h}_1)]m + \sqrt{\frac{m \log(m/\beta)}{2}}\Big),$$

$$K = O\Big(\frac{\log(mT/\min(\delta,\beta))\sqrt{T\log(1/\delta)}}{\epsilon}\Big).$$

Let $\hat{h}^S$ be the output of Algorithm 6 that uses Algorithm 5 for privacy aggregation. With probability at least $1-\beta$ (over the randomness of the algorithm and the randomness of all data points drawn i.i.d.), we have

$$\texttt{Err}(\hat{h}^S) \leq \tilde{O}\Big(\frac{d^2 m\log(1/\delta)}{n^2\epsilon^2} + \sqrt{\frac{d}{m}}\Big)$$

for the realizable case, and

$$\texttt{Err}(\hat{h}^S) \leq 13\texttt{Err}(h^*) + \tilde{O}\Big(\frac{m^{1/3}d^{2/3}}{n^{2/3}\epsilon^{2/3}} + \sqrt{\frac{d}{m}}\Big)$$

for the agnostic case [1].

We provide a self-contained proof of this result in Section 4.7.

Remark 4.3.6 (Error bounds when m is sufficiently large) *Notice that we do not have to label all public data, so when we have a large number of public data, we can afford to choose m to be smaller so as to minimize the bound. That gives us a $\tilde{O}(\frac{d}{n^{2/3}\epsilon^{2/3}})$ error bound for the realizable case and a $O(\texttt{Err}(h^*)) + \tilde{O}(\frac{d^{3/5}}{n^{2/5}\epsilon^{2/5}})$ error bound for the agnostic case* [2].

[1] The numerical constant 13 might be improvable (and it is indeed worse than the result stated in [88]), though we decide to present this for the simplicity of the proof.

[2] These correspond to the $\tilde{O}((d/\alpha)^{3/2})$ sample complexity bound in Theorem 4.6 of [106] for realizable PAC learning for error α; and the $\tilde{O}(d^{3/2}/\alpha^{5/2})$ sample complexity bound in Theorem 4.7 of [106] for agnostic PAC learning with error $O(\alpha + \texttt{Err}(h^*))$. The privacy parameter ϵ is taken as a constant in these results.

Algorithm 7 Disagreement-Based Active Learning [93]

Input: A "data stream" $x_1, x_2, ...$ sampled i.i.d. from distribution $\mathcal{D}$. A hypothesis class $\mathcal{H}$. An on-demand "labeling service" that outputs label $y_i \sim P(y|x = x_i)$ when requested at time i. Parameter ℓ, m, γ.

Initialize the version space $V \leftarrow \mathcal{H}$.
Initialize the selected dataset $Q \leftarrow \varnothing$.
Initialize "Current Output" to be any $h \in \mathcal{H}$.
Initialize "Counter" $c \leftarrow 0$.
for $j \in [m]$ **do**
  if $x_j \in \texttt{DIS}(V)$ **then**
    "Request for label" for x_j and get back y_j from the "labeling service".
    Update $Q \leftarrow Q \cup \{(x_j, y_j)\}$.
    $c \leftarrow c + 1$.
  end if
  if $\log_2(j) \in \mathbb{N}$ **then**
    Update $V \leftarrow \{h \in V : (\texttt{Err}_Q(h) - \min_{g \in V} \texttt{Err}_Q(g))|Q| \leq U(j, \gamma_j)j\}$, where $U(j, \gamma_j) = c'(d\log(\theta(d/j)) + \log(1/\gamma_j))/j + c'\sqrt{\texttt{Err}(h^*)(d\log(\theta(\texttt{Err}(h^*)) + \log(1/\gamma_j))/j}$, c' is a constant, and $\gamma_j = \gamma/(\log_2(2j))^2$.
    Set "Current Output" to be any $h \in V$.
  end if
  if $c \geq \ell$ **then**
    Break.
  end if
end for

Output: Return "Current Output".

4.3.4 Disagreement-Based Active Learning

We adopt the disagreement-based active learning algorithm that comes with strong learning bounds (see, e.g., an excellent treatment of the subject in [93]). The exact algorithm, described in Algorithm 7, keeps updating a subset of the hypothesis class $\mathcal{H}$ called a *version space* by collecting labels only from those data points from a certain *region of disagreement* and eliminates candidate hypothesis that are certifiably suboptimal.

Definition 4.3.7 (Region of disagreement [93]) *For a given hypothesis class $\mathcal{H}$, its region of disagreement is defined as a set of data points over which there exists two hypotheses disagreeing with each other,*

$$\texttt{DIS}(\mathcal{H}) = \{x \in \mathcal{X} : \exists h, g \in \mathcal{H} \text{ s.t. } h(x) \neq g(x)\}.$$

Region of disagreement is the key concept of the disagreement-based active learning algorithm. It captures the uncertainty region of data points for the current version space. The algorithm is fed a sequence of data points and runs in the online fashion, whenever there exists a data point in this region, its label will be queried. Then any *bad* hypotheses will be removed from the version space.

The algorithm, as it is written is not directly implementable, as it represents the version spaces explicitly, but there are practical implementations that avoids explicitly representing the versions spaces by a reduction to supervised learning oracles. In our experiments, we implement the PATE-ASQ algorithm and show it works well in practice while no explicit region of disagreement is maintained.

4.4 Main Results

In Section 4.4.1 and 4.4.2, we present a more refined theoretical analysis of PATE-PSQ (Algorithm 6) that uses SVT-based PATE (Algorithm 5) as the subroutine. Our results provide stronger learning bounds and new theoretical insights under various settings. In Section 4.4.3, we propose a new active learning based method and show that we can obtain qualitatively the same theoretical gain while using the simpler (an often more practical) Gaussian mechanism-based PATE (Algorithm 4) as the subroutine. For comparison, we also include an analysis of standard PATE (with Gaussian mechanism) in Section 4.7. Table 4.1 summarizes these technical results.

4.4.1 Improved Learning Bounds under TNC

Recall that our motivation is to analyze PATE in the cases when the best classifier does not achieve 0 error and that existing bound presented in Theorem 4.3.5 is vacuous if $\texttt{Err}(h^*) > 1/26$. The error bound of $\hat{h}^S$ does not match the performance of h^* even as $m, n \to \infty$ and even if we output the voted labels without adding noise. This does not explain the empirical performance of Algorithm 6 reported in [85, 86] which demonstrates that the retrained classifier from PATE could get quite close to the best non-private baselines even if the latter are far from being perfect. For instance, on Adult dataset and SVHN dataset, the non-private baselines have accuracy 85% and 92.8% and PATE achieves 83.7% and 91.6% respectively.

To under stand how PATE works in the regime where the best classifier h^* obeys that $\texttt{Err}(h^*) > 0$, we introduce a large family of learning problems that satisfy the so-called Tsybakov Noise Condition (TNC), under which we show that PATE is consistent with fast rates. To understand TNC, we need to introduce a few more notations. Let label $y \in \{0, 1\}$ and define the regression function $r(x) = \mathbb{E}[y|x]$. The Tsybakov noise

condition is defined in terms of the distribution of $r(x)$.

Definition 4.4.1 (Tsybakov noise condition) *The joint distribution of the data* (x, y) *satisfies the Tsybakov noise condition with parameter* τ *if there exists a universal constant* $C > 0$ *such that for all* $t \geq 0$

$$\mathbb{P}(|r(x)| \leq t) \leq Ct^{\frac{\tau}{1-\tau}}.$$

Note that when $r(x) = 0.5$, the label is purely random and when $r(x) = 0$ or 1, y is a deterministic function of x. The Tsybakov noise condition essentially is reasonable "low noise" condition that does not require a uniform lower bound of $|r(x)|$ for all x. When the label-noise is bounded for all x, e.g., when $y = h^*(x)$ with probability 0.6 and $1 - h^*(x)$ with probability 0.4, then the Tsybakov noise condition holds with $\tau = 1$. The case when $\tau = 1$ is also known as the *Massart noise condition* or *bounded noise condition* in the statistical learning literature.

For our purpose, it is more convenient to work with the following equivalent definition of TNC, which is equivalent to Definition 4.4.1 (see a proof from [107, Definition 7]).

Lemma 4.4.2 (Equivalent definition of TNC) *We say that a distribution of* (x, y) *satisfies the Tsybakov noise condition with parameter* $\tau \in [0, 1]$ if and only if *there exists* $\eta \in [1, \infty)$ *such that, for every labeling function* h*,*

$$\mathrm{Dis}(h, h_{\mathrm{Bayes}}) \leq \eta(\mathtt{Err}(h) - \mathtt{Err}(h_{\mathrm{Bayes}}))^{\tau}. \tag{4.2}$$

where $h_{\mathrm{Bayes}}(x) = \mathbb{1}(r(x) > 0.5)$ *is the Bayes optimal classifier.*

In the remainder of this chapter, we make the assumption that the Bayes optimal classifier $h_{\mathrm{Bayes}} \in \mathcal{H}$ and works with the slightly weaker condition that requires (4.2) to hold only

for $h \in \mathcal{H}$ and that we replace h_{Bayes} by the optimal classifier $h^* \in \mathcal{H}$ [3].

We emphasize that the Tsybakov noise condition is not our invention. It has a long history from statistical learning theory to interpolate between the realizable setting and the agnostic setting. Specifically, problems satisfying TNC admit fast rates. For $\tau \in [0, 1]$, the empirical risk minimizer achieves an excess risk of $O(1/n^{1/(2-\tau)})$, which clearly interpolates the realizable case of $O(1/n)$ and the agnostic case of $O(1/\sqrt{n})$.

Next, we give a novel analysis of Algorithm 6 under TNC. The analysis is simple but revealing, as it not only avoids the strong assumption that requires $\texttt{Err}(h^*)$ to be close to 0, but also achieves a family of fast rates which significantly improves the sample complexity of PATE learning even for the realizable setting.

Theorem 4.4.3 (Utility guarantee of Algorithm 6 under TNC) *Assume the data distribution $\mathcal{D}$ and the hypothesis class $\mathcal{H}$ obey the Tsybakov noise condition with parameter τ. Then Algorithm 6 with*

$$T = \tilde{O}\Big(\Big(\frac{m^{2-\tau} d^\tau}{n^\tau \epsilon^\tau}\Big)^{\frac{2}{4-3\tau}}\Big),$$

$$K = O\Big(\frac{\log(mT/\min(\delta, \beta))\sqrt{T\log(1/\delta)}}{\epsilon}\Big),$$

obeys that with probability at least $1 - \beta$:

$$\texttt{Err}(\hat{h}^S) \leq \texttt{Err}(h^*) + \tilde{O}\left(\frac{d}{m} + \left(\frac{md^2}{n^2\epsilon^2}\right)^{\frac{\tau}{4-3\tau}}\right).$$

Remark 4.4.4 (Bounded noise case) *When $\tau = 1$, the Tsybakov noise condition is*

[3] This slightly different condition, that requires (4.2) to hold only for $h \in \mathcal{H}$ but with h_{Bayes} replaced by the optimal classifier h^* (without assuming that $h^* = h_{\text{Bayes}}$) is all we need. This is formally referred to as the Bernstein class condition by [93]. Very confusingly, when the Tsybakov noise condition is being referred to in more recent literature, it is in fact the Bernstein class condition — a slightly weaker but more opaque definition about both the hypothesis class $\mathcal{H}$ and the data generating distribution.

implied by the bounded noise assumption, a.k.a., Massart noise condition, where the labels are generated by the Bayes optimal classifier h^* and then toggled with a fixed probability less than 0.5. *Theorem 4.4.3 implies that the excess risk is bounded by $\tilde{O}(\frac{d^2m}{n^2\epsilon^2} + \frac{d}{m})$, with $K = \tilde{O}(\frac{dm}{n\epsilon^2})$, which implies a sample complexity upper bound of $\tilde{O}(\frac{d^{3/2}}{\alpha\epsilon})$ private data points and $\tilde{O}(d/\alpha)$ public data points. The results improve over the sample complexity bound from [88] in the stronger realizable setting from $\tilde{O}(\frac{d^{3/2}}{\alpha^{3/2}\epsilon})$ and $\tilde{O}(d/\alpha^2)$ to $\tilde{O}(\frac{d^{3/2}}{\alpha\epsilon})$ and $\tilde{O}(d/\alpha)$ respectively in the private and public data.*

Remark 4.4.5 (Optimal choice of m) *The upper bound above can be minimized by choosing $m^* = (d^{4-5\tau}n^{2\tau}\epsilon^{2\tau})^{\frac{1}{4-2\tau}}$. When number of available public data points $m \geq m^*$, then m is not a limiting factor and we should subsample these data points. When $m < m^*$, then d/m is the leading factor, we should use all m data points.*

There are two key observations behind the improvement. First, the teacher classifiers do not have to agree on the labels y as in Lemma 4.3.4; all they have to do is to agree on something for the majority of the data points. Conveniently, the Tsybakov noise condition implies that the teacher classifiers agree on the Bayes optimal classifier h^*. Second, when the teachers agree on h^*, the synthetic learning problem with the privately released pseudo-labels is nearly realizable. These intuitions can be formalized with a few lemmas, which will be used in the proof of Theorem 4.4.3.

Lemma 4.4.6 (Performance of teacher classifer w.r.t. h^*) *With probability $1-\gamma$ over the training data of $\hat{h}_1, ..., \hat{h}_K$, assume $h^* \in \mathcal{H}$ is the Bayes optimal classifier and Tsybakov noise condition with parameter τ, then there is a universal constant C such that for all $k = 1, 2, 3, ..., K$*

$$\mathrm{Dis}(\hat{h}_k, h^*) \leq C\Big(\frac{dK\log(n/d) + \log(K/\gamma)}{n}\Big)^{\frac{\tau}{2-\tau}}.$$

Proof: By the equivalent definition of the Tsybakov noise condition and then the learning bounds under TNC (Lemma A.0.9),

$$\mathrm{Dis}(\hat{h}_k, h^*) \leq \eta(\mathtt{Err}(\hat{h}_k, h^*) - \mathtt{Err}(h^*))^\tau \leq C\Big(\frac{dK\log(n/d) + \log(K/\gamma)}{n}\Big)^{\frac{\tau}{2-\tau}}.$$

■

Lemma 4.4.7 (Total number of mistakes made by one teacher) *Under the condition of Lemma 4.4.6, with probability $1-\gamma$, for all $k = 1, 2, ..., K$ the total number of mistakes made by one teacher classifier $\hat{h}_k$ with respect to h^* can be bounded as:*

$$\sum_{j=1}^{m} \mathbb{1}(\hat{h}_k(x_j) \neq h^*(x_j)) \leq O\bigg(\max\bigg\{m\mathrm{Dis}(\hat{h}_k, h^*), \log\Big(\frac{K}{\gamma}\Big)\bigg\}\bigg).$$

Proof: Number of mistakes made by $\hat{h}_k$ with respect to h^* is the empirical disagreement between $\hat{h}_k$ and h^* on m data points, therefore, by Bernstein's inequality (Lemma A.0.6),

$$\begin{aligned}\sum_{j=1}^{m} \mathbb{1}(\hat{h}_k(x_j) \neq h^*(x_j)) &\leq O\bigg(m\mathrm{Dis}(\hat{h}_k, h^*) + \sqrt{m\mathrm{Dis}(\hat{h}_k, h^*)\log\Big(\frac{K}{\gamma}\Big)} + \log\Big(\frac{K}{\gamma}\Big)\bigg) \\ &\leq O\bigg(\max\bigg\{m\mathrm{Dis}(\hat{h}_k, h^*), \log\Big(\frac{K}{\gamma}\Big)\bigg\}\bigg).\end{aligned}$$

■

Using the above two lemmas we establish a bound on the number of examples where the differentially privately released labels differ from the prediction of h^*.

Lemma 4.4.8 (Total queries and cut-off budget) *Let Algorithm 6 be run with the number of teachers K and the cut-off parameter T chosen according to Theorem 4.4.3. Assume the conditions of Lemma 4.4.6. Then with high probability ($\geq 1-\beta$ over the*

random coins of Algorithm 6 alone and conditioning on the high probability events of Lemma 4.4.6 and Lemma 4.4.7), Algorithm 6 finishes all m queries without exhausting the cut-off budget and that

$$\sum_{j=1}^{m} \mathbb{1}(\hat{h}_j^{\texttt{priv}} \neq h^*(x_j)) \leq T.$$

The $\tilde{O}$ notation in the choice of K and T hides polynomial factors of $\log(K/\gamma), \log(m/\beta)$ where γ is from Lemma 4.4.6 and 4.4.7.

Proof: Denote the bound from Lemma 4.4.7 by B. By the same Pigeon hole principle argument as in Lemma 4.3.4 (but with y replaced by h^*), we have that the number of queries that have margin smaller than $K/6$ is at most $3B = O(\max\{m\text{Dis}(\hat{h}_k, h^*), \log(K/\gamma)\})$. The choice of K ensures that with high probability, over the Laplace random variables in Algorithm 5, in at least $m - 3B$ queries where the answer $\hat{y}_j = h^*(x_j)$, i.e.,

$$\sum_{j=1}^{m} \mathbb{1}(\hat{h}_j^{\texttt{priv}} \neq h^*(x_j)) \leq 3B := T.$$

■

Now we are ready to put everything together and prove Theorem 4.4.3. *Proof:* [Proof of Theorem 4.4.3] Denote $\tilde{h} = \text{argmin}_{h \in \mathcal{H}} \widehat{\text{Dis}}(h, h^*)$ where $\widehat{\text{Dis}}$ is the empirical average of the disagreements over the data points that students have[4]. By the triangular

[4]Note that in this case we could take $\tilde{h} = h^*$ since $h^* \in \mathcal{H}$. We are defining this more generally so later we can substitute h^* with other label vector that are not necessarily generated by any hypothesis in $\mathcal{H}$.

inequality of the $0-1$ error,

$$\begin{aligned}\texttt{Err}(\hat{h}^S) - \texttt{Err}(h^*) &\leq \mathrm{Dis}(\hat{h}^S, h^*) \\ &\leq \widehat{\mathrm{Dis}}(\hat{h}^S, h^*) + 2\sqrt{\frac{(d+\log(4/\gamma))\widehat{\mathrm{Dis}}(\hat{h}^S, h^*)}{m}} + \frac{4(d+\log(4/\gamma))}{m} \\ &\leq 2\widehat{\mathrm{Dis}}(\hat{h}^S, h^*) + \frac{5(d+\log(4/\gamma))}{m}, \end{aligned} \tag{4.3}$$

where the second line follows from the uniform Bernstein's inequality — apply the first statement Lemma A.0.7 in Appendix A with $z = h^*(x)$ and the third line is due to $a + 2\sqrt{ab} + b \leq 2a + 2b$ for non-negative a, b.

By the triangular inequality, we have $\widehat{\mathrm{Dis}}(\hat{h}^S, h^*) \leq \widehat{\mathrm{Dis}}(\hat{h}^S, \hat{h}^{\texttt{priv}}) + \widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h^*)$, therefore

$$\begin{aligned}(4.3) &\leq 2\widehat{\mathrm{Dis}}(\hat{h}^S, \hat{h}^{\texttt{priv}}) + 2\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h^*) + \frac{5(d+\log(4/\gamma))}{m} \\ &\leq 2\widehat{\mathrm{Dis}}(\tilde{h}, \hat{h}^{\texttt{priv}}) + 2\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h^*) + \frac{5(d+\log(4/\gamma))}{m} \\ &\leq 2\widehat{\mathrm{Dis}}(\tilde{h}, h^*) + 4\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h^*) + \frac{5(d+\log(4/\gamma))}{m} \\ &= 4\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h^*) + \frac{5(d+\log(4/\gamma))}{m}. \end{aligned}$$

In the second line, we applied the fact that $\hat{h}^S$ is the minimizer of $\widehat{\mathrm{Dis}}(h, \hat{h}^{\texttt{priv}})$; in the third line, we applied triangular inequality again and the last line is true because $\widehat{\mathrm{Dis}}(\tilde{h}, h^*) = 0$ since $\tilde{h}$ is the minimizer and that $h^* \in \mathcal{H}$.

Recall that T is the unstable cutoff in Algorithm 6. The proof completes by invoking Lemma 4.4.8 which shows that the choice of T is appropriate such that $\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h^*) \leq T/m$ with high probability. ■

In the light of the above analysis, it is clear that the improvement from our analysis under TNC are two-folds: (1) We worked with the disagreement with respect to h^* rather

than y. (2) We used a uniform Bernstein bound rather than a uniform Hoeffding bound that leads to the faster rate in terms of the number of public data points needed.

Remark 4.4.9 (Reduction to ERM) *The main challenge in the proof is to appropriately take care of $\hat{h}^{\texttt{priv}}$. Although we are denoting it as a classifier, it is in fact a vector that is defined only on $x_1, ..., x_m$ rather than a general classifier that can take any input x. Since we are using the SVT-based Algorithm 5, $\hat{h}^{\texttt{priv}}$ is only well-defined for the student dataset. Moreover, these privately released "pseudo-labels" are not independent, which makes it infeasible to invoke a generic learning bound such as Lemma A.0.8. Our solution is to work with the empirical risk minimizer (ERM, rather than a generic PAC learner as a blackbox) and use uniform convergence (Lemma A.0.7) directly. This is without loss of generality because all learnable problems are learnable by (asymptotic) ERM [108, 109].*

4.4.2 Challenges and New Bounds under Agnostic Setting

In this section, we present a more refined analysis of the agnostic setting. We first argue that agnostic learning with Algorithm 6 will not be consistent in general and competing against the best classifier in $\mathcal{H}$ seems not the right comparator. The form of the pseudo-labels mandate that $\hat{h}^S$ is aiming to fit a labeling function that is inherently a voting classifier. The literature on ensemble methods has taught us that the voting classifier is qualitatively different from the individual voters. In particular, the error rate of majority voting classifier can be significantly better, about the same, or significantly worse than the average error rate of the individual voters. We illustrate this with two examples.

Example 4.4.10 (Voting fails) *Consider a uniform distribution on $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$ and that the corresponding label $\mathbb{P}(y = 1) = 1$. Let the hypothesis class be $\mathcal{H} = \{h_1, h_2, h_3\}$ whose evaluation on $\mathcal{X}$ are given in Figure 4.1. Check that the classification error of all*

three classifiers is 0.5. *Also note that the empirical risk minimizer* $\hat{h}$ *will be a uniform distribution over* h_1, h_2, h_3. *The majority voting classifiers, learned with iid data sets, will perform significantly worse and converge to a classification error of* 0.75 *exponentially quickly as the number of classifiers* K *goes to* ∞.

	x_1	x_2	x_3	x_4	Error
y	1	1	1	1	0
h_1	1	1	0	0	0.5
h_2	1	0	1	0	0.5
h_3	1	0	0	1	0.5
$\hat{h}^{\mathtt{agg}}$	1	0	0	0	0.75

Figure 4.1: An example where majority voting classifier is significantly worse than the best classifier in $\mathcal{H}$.

This example illustrates that the PATE framework cannot consistently learn a VC-class in the agnostic setting in general. On a positive note, there are also cases where the majority voting classifier boosts the classification accuracy significantly, such as the following example.

Example 4.4.11 (Voting wins) *If* $\mathbb{P}[\hat{h}(x) \neq y|x] \leq 0.5-\xi$, *where* ξ *is a small constant, for all* $x \in \mathcal{X}$, *then by Hoeffding's inequality,*

$$\mathbb{P}[\hat{h}^{\mathtt{agg}}(x) \neq y|x] = \mathbb{P}\Big[\sum_{k=1}^{K} \mathbb{1}(\hat{h}_k(x) \neq y) \geq \frac{k}{2}\Big|x\Big] \leq e^{-2K\xi^2}.$$

Thus the error goes to 0 *exponentially as* $K \to \infty$.

These cases call for an alternative distribution-dependent theory of learning that characterizes the performance of Algorithm 6 more accurately.

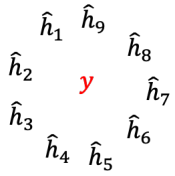
Next, we propose two changes to the learning paradigms. First, we need to go beyond

$\mathcal{H}$ and compare with the following infinite ensemble classifier

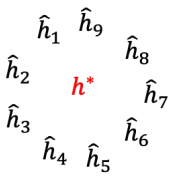
$$h_\infty^{\text{agg}}(x) := \mathbb{1}\left(\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{k}\hat{h}_k(x)\middle|x\right] \geq \frac{1}{2}\right) = \mathbb{1}\left(\mathbb{E}[\hat{h}_1(x)|x] \geq \frac{1}{2}\right).$$

The classifier outputs the majority voting result of infinitely many independent teachers, each trained on n/K i.i.d. data points. As discussed earlier, this classifier can be better or worse than a single classifier $\hat{h}_1$ that takes n/K data points, $\hat{h}$ that trains on all n data points or h^* that is the optimal classifier in $\mathcal{H}$. Note that this classifier also changes as n/K gets larger.

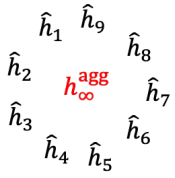
Considering different centers for teacher classifiers to agree on is one of the key ideas of this chapter. Figure 4.2 shows three kinds of centers for teachers $\hat{h}_1, \hat{h}_2, ..., \hat{h}_9$ to agree on. In [88], the center is the true label y in the realizable setting. In Section 4.4.1 under TNC, we analyze the performance of PATE-PSQ, where the center is the best hypothesis h^*. Now we are interested in the new center h_∞^{agg} for teachers to agree on.



(a) True label y is the center for in realizable setting.



(b) Best hypothesis h^* is the center under TNC.



(c) h_∞^{agg} is our new construction for agnostic setting.

Figure 4.2: Centers for teachers $\hat{h}_1, \hat{h}_2, ..., \hat{h}_9$ to agree on.

Second, we define the *expected margin* for a classifier $\hat{h}_1$ trained on n i.i.d. samples to be

$$\Delta_n(x) := \left|\mathbb{E}[\hat{h}_1(x)|x] - \frac{1}{2}\right|. \tag{4.4}$$

This quantity captures for a fixed $x \in \mathcal{X}$, how likely the teachers will agree. For a

fixed learning problem $\mathcal{H}, \mathcal{D}$ and the number of i.i.d. data points $\hat{h}_1$ is trained upon, the expected margin is a function of x alone. The larger $\Delta_{n/K}(x)$ is, the more likely that the ensemble of K teachers agree on a prediction in $\mathcal{Y}$ with high-confidence. Note that unlike in Example 4.4.11, we do not require the teachers to agree on y. Instead, it measures the extent to which they agree on $h_\infty^{\mathtt{agg}}(x)$, which could be any label.

When the expected margin is bounded away from 0 for x, then the voting classifier outputs $h_\infty^{\mathtt{agg}}(x)$ with probability converging exponentially to 1 as K gets larger. On the technical level, this definition allows us to *decouple* the stability analysis and accuracy of PATE as the latter relies on how good $h_\infty^{\mathtt{agg}}$ is.

Definition 4.4.12 (Approximate high margin) *We say that a learning problem with n i.i.d. samples satisfy (ν, ξ)-approximate high-margin condition if $\mathbb{P}_{x\sim\mathcal{D}}[\Delta_n(x) > \xi] \leq \nu$.*

This definition says that with high probability, except for $O(\nu m)$ data points, all other data points in the public dataset have an expected margin of at least ξ. Observe that every learning problem has ξ that increases from 0 to 0.5 as we vary ν from 0 to 1. The realizability assumption and the Tsybakov noise condition that we considered up to this point imply upper bounds of ν at fixed ξ (see more details in Remark 4.4.16).

The following proposition shows that when a problem is approximate high-margin, there are choices T and K under which the SVT-based PATE provably labels almost all data points with the output of $h_\infty^{\mathtt{agg}}$.

Proposition 4.4.13 *Assume the learning problem with n/K i.i.d. data points satisfies (ν, ξ)-approximate high-margin condition. Let Algorithm 5 be instantiated with parameters*

$$T \geq \nu m + \sqrt{2\nu m \log\left(\frac{3}{\gamma}\right)} + \frac{2}{3}\log\left(\frac{3}{\gamma}\right),$$

$$K \geq \max\left\{\frac{2\log(3m/\gamma)}{\xi^2}, \frac{3\lambda\big(\log(4m/\delta)+\log(3m/\gamma)\big)}{\xi}\right\},^{5}$$

then with high probability (over the randomness of the n i.i.d. samples of the private dataset, m i.i.d. samples of the public dataset, and that of the randomized algorithm), Algorithm 5 finishes all m rounds and the output is the same as $h_\infty^{\mathtt{agg}}(x_i)$ for all but T of the $i \in [m]$.

This proposition provides the utility guarantee to Algorithm 5 and generalizes Lemma 4.4.8 from fixing $\xi = 1/6$ into allowing much smaller ξ at a cost of increasing ν.

Next, we state the learning bounds under the approximate-high margin condition.

Theorem 4.4.14 *Assume the learning problem with n/K i.i.d. data points satisfies (ν,ξ)-approximate high-margin condition and let K, T be chosen according to Proposition 4.4.13, furthermore assume that the privacy parameter of choice $\epsilon \leq \log(2/\delta)$, then the output classifier $\hat{h}^S$ of Algorithm 6 in the agnostic setting satisfies that with probability $\geq 1-2\gamma$,*

$$\begin{aligned}\mathtt{Err}(\hat{h}^S) - \mathtt{Err}(h_\infty^{\mathtt{agg}}) &\leq \min_{h\in\mathcal{H}} \mathrm{Dis}(h, h_\infty^{\mathtt{agg}}) + \frac{2T}{m} + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\ &\leq \min_{h\in\mathcal{H}} \mathrm{Dis}(h, h_\infty^{\mathtt{agg}}) + 2\nu + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big).\end{aligned}$$

The voting classifier $\hat{h}^{\mathtt{agg}}$ is usually not in the original hypothesis class $\mathcal{H}$, so we can take a wider view of the hypothesis class and define the voting hypothesis space $\mathrm{Vote}(\mathcal{H})$ where the learning problem becomes realizable. Note if the VC dimension of $\mathcal{H}$ is d, then the VC dimension of $\mathrm{Vote}_K(\mathcal{H}) \leq Kd$. In practice, this suggests using ensemble methods such as AdaBoost for K iterations.

Theorem 4.4.15 *Under the same assumption of Theorem 4.4.14, suppose we train an*

[5] $\lambda = (\sqrt{2T(\epsilon+\log(2/\delta))} + \sqrt{2T\log(2/\delta)})/\epsilon$ according to Algorithm 5.

ensemble classifier within the voting hypothesis space $\mathrm{Vote}_K(\mathcal{H})$ *in the student domain, then the output classifier* $\hat{h}^S$ *of Algorithm 6 in the agnostic setting satisfies that with probability* $\geq 1 - 2\gamma$,

$$\mathtt{Err}(\hat{h}^S) - \mathtt{Err}(h_\infty^{\mathtt{agg}}) \leq \frac{4T}{m} + \frac{5(Kd + \log(4/\gamma))}{m} = \tilde{O}\Big(\nu + \frac{\log(4/\gamma)}{m} + \frac{d\sqrt{\nu}}{\xi\sqrt{m}}\Big).$$

Remark 4.4.16 *Whether the bounds in Theorem 4.4.14 and 4.4.15 will vanish as* $m, n \to \infty$ *depends strongly on how parameter* ν *and* ξ *change as* n/K *gets larger. Intuitively, if the learner converges to a single classifier* h^**, as in the realizable case or under TNC, then we can show that the learning problem satisfy* (ν, ξ)*-approximate high-margin condition with* $\xi = 1/6$ *and* $\nu \leq \tilde{O}((dK/n)^{\frac{\tau}{2-\tau}})$*. Substituting this quantities into Theorem 4.4.14 and using the fact that* ν *also bounds the disagreement between* h^* *and* $h_\infty^{\mathtt{agg}}$ *allows us obtain a bound that vanishes as* n *gets larger. More generally, in the agnostic case, it is reasonable to assume that the "teachers" will get* more confident *in their individual prediction for most data points as* $n/K \to \infty$*. We argue this is a more modest requirement than requiring the "teachers" to get* more accurate*.*

4.4.3 PATE with Active Student Queries

In previous subsections, we have proved stronger learning bounds for PATE framework under TNC and in agnostic setting. However, all these results are based on the variants of PATE that aim at *passively* releasing *almost all* student queries. In this section we address the following question:

Can we do even better if we cherry-pick queries to label?

The hope is that this allows us to spend privacy budget only on those queries that add new information for the interest of training a classifier, hence resulting in a more

favorable privacy-utility tradeoff. Without privacy constraints, this problem is known as active learning and it is often possible to save exponentially in the number of labels needed comparing to the passive learning model.

In Algorithm 8, we propose a new algorithm called PATE with Active Student Queries (PATE-ASQ) which uses the disagreement-based active learning algorithm (Algorithm 7) as the subroutine. Then we provide its utility guarantee.

Algorithm 8 PATE-ASQ

Input: Labeled private teacher dataset D^T, unlabeled public student dataset D^S, privacy parameters $\epsilon, \delta > 0$, number of splits K, maximum number of queries ℓ, failure probability γ.

1: Randomly and evenly split the teacher dataset D^T into K parts $D_k^T \subseteq D^T$ where $k \in [K]$
2: Train K classifiers $\hat{h}_k \in \mathcal{H}$, one from each part D_k^T.
3: Declare "Labeling Service" $\leftarrow$ Algorithm 4 with $\hat{h}_1, ..., \hat{h}_K$, ℓ, ϵ, δ, with an unspecified "nature".
4: Initiate an active learning oracle (e.g., Algorithm 7) with an iterator over D^S being the "data stream", hypothesis class $\mathcal{H}$, failure probability γ. Set the "labeling service" to be Algorithm 4 with parameter $\hat{h}_1, ..., \hat{h}_K$, ℓ, ϵ, δ, and set the "nature" to be the "request for label" calls in the active learning oracle.
5: Set $\hat{h}^S$ to be the "current output" from active learning oracle.

Output: Return $\hat{h}^S$.

Theorem 4.4.17 (Utility guarantee of Algorithm 8) *With probability at least* $1 - \gamma$*, there exists universal constants* C_1, C_2 *such that for all*

$$\alpha \geq C_1 \max\left\{\eta^{\frac{2}{2-\tau}}\left(\frac{dK\log(n/d) + \log(2K/\gamma)}{n}\right)^{\frac{\tau}{2-\tau}}, \frac{d\log((m+n)/d) + \log(2/\gamma)}{m}\right\},$$

the output $\hat{h}^S$ *of Algorithm 8 with parameter* ℓ, K *satisfying*

$$\ell = C_2\theta(\alpha)\Big(1 + \log\big(\frac{1}{\alpha}\big)\Big)\left(d\log(\theta(\alpha)) + \log\left(\frac{\log(1/\alpha)}{\gamma/2}\right)\right)$$

$$K = \frac{6\sqrt{\log(2n)}(\sqrt{\ell \log(1/\delta)} + \sqrt{\ell \log(1/\delta) + \epsilon\ell})}{\epsilon}$$

obeys that

$$\texttt{Err}(\hat{h}^S) - \texttt{Err}(h^*) \leq \alpha.$$

Specifically, when we choose

$$\alpha = C_1 \max\left\{\eta^{\frac{2}{2-\tau}}\left(\frac{dK\log(n/d) + \log(2K/\gamma)}{n}\right)^{\frac{\tau}{2-\tau}}, \frac{d\log((m+n)/d) + \log(2/\gamma)}{m}\right\},$$

and also $\epsilon \leq \log(1/\delta)$*, then it follows that*

$$\texttt{Err}(\hat{h}^S) - \texttt{Err}(h^*) = \tilde{O}\left(\max\left\{\left(\frac{d^{1.5}\sqrt{\theta(\alpha)\log(1/\delta)}}{n\epsilon}\right)^{\frac{\tau}{2-\tau}}, \frac{d}{m}\right\}\right),$$

where $\tilde{O}$ *hides logarithmic factors in* $m, n, 1/\gamma$*.*

Remark 4.4.18 *The bound above resembles the learning bound we obtain using the passive student queries with Algorithm 5 as the privacy procedure, except for the additional dependence on the disagreement coefficients. Interestingly, active learning achieves this bound without using the sophisticated (and often not practical) algorithmic components from DP, such as sparse sector technique to save privacy losses. Instead, we can get away with using simple Gaussian mechanism as in Algorithm 4.*

Remark 4.4.19 (Blackbox reduction, revisited) *In contrary to our discussion in Remark 4.4.9, notice that we are using Algorithm 4 instead of Algorithm 5 as the labeling services, which allows us to reduce to any learner as a blackbox. This makes it possible to state formally results even for deep neural networks or other family of methods where obtaining ERM is hard but learning is conjectured to be easy in theory and in practice.*

Remark 4.4.20 (Relationships between SVT and active learning) *There is an in-*

triguing analogy between the Algorithm 5 which simply labels all queries with an advanced DP mechanism and Algorithm 4 which uses active learning with a simple DP mechanism. On a high level, both approaches are doing selection. Active learning selects those queries that are near the decision boundary to be informative for learning; the sparse-vector-technique approach essentially *selects those queries that are not stable to spend privacy budget on.*

One curious question is whether the two sets of selected data points are substantially overlapping. If not, then we might be able to combine the two and achieve even better private-utility tradeoff.

4.5 Experiments

In this section, we present our empirical studies of PATE-PSQ and PATE-ASQ algorithms. Section 4.5.1 describes how we set up our experiments, and Section 4.5.2 show our results.

4.5.1 Experimental Settings

Algorithms compared. We focus on comparing the classification accuracy of the passive and active learning versions of PATE on a holdout test set ("Utility") when both algorithms are calibrated to the same privacy budget ϵ ("privacy"). To set baselines, we also compare them with non-private versions of them (no noise added to the votes, or $\epsilon = +\infty$), denoted by PATE-PSQ-NP and PATE-ASQ-NP. We remark that the PATE-PSQ we implement is the Gaussian mechanism version [86]. While we have shown that it has higher *asymptotic* sample complexity comparing to the more advanced version based on SVT [88] (Section 4.4.1), we found that the Gaussian mechanism version performs better for the realistically-sized datasets that we considered. Linear models are used

for all of these algorithms for simplicity. For active learning, we follow the practical implementation of the disagreement-based active learning by [110], which does not require the learner to explicitly maintain the (exponentially large) region of disagreement.

Datasets. We do our experiments on three binary classification datasets, mushroom, a9a, and real-sim. All of them are obtained from LIBSVM dataset website [6]. See Table 4.3 for the statistics of them. If a dataset had been previously split into training and testing parts, we combine them together and record the total number of all data points. For all datasets, 80% of all data points are randomly selected to be considered private and used to train teacher classifiers. 2% of all data points are randomly selected as public student unlabeled data points. The remaining 18% data points are reserved for testing. We repeat these random selection processes for 30 times.

Table 4.3: Statistics of datasets.

Dataset	**# All**	**# Train**	**# Unlabeled**	**# Budget**	**# Test**	**# Dimension**
mushroom	$8,124$	$6,499$	163	49	$1,462$	112
a9a	$48,842$	$39,073$	977	293	$8,792$	123
real-sim	$72,309$	$57,847$	$1,447$	434	$13,015$	$20,958$

Parameter settings. Number of teachers K is set on all datasets so that each teacher classifier gets trained with approximately 100 data points. 30% of student unlabeled data points are set as the total budget of queries for PATE-ASQ and PATE-ASQ-NP. See Table 4.3. $\epsilon = 0.5, 1.0, 2.0$ and $\delta = 1/n$ are set as privacy parameters for all datasets, where n is number of private teacher data points. All privacy accounting and calibration are conducted via AutoDP [111], and the tight analytical calibration and composition of Gaussian mechanisms are due to [112].

[6] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Privacy loss vs. privacy budget. Besides the privacy budget parameter ϵ that the algorithms receive as an input, it is often the case that the active learning algorithm halts before exhausting the query budget of (30% of the total number of unlabeled data points). Therefore the privacy loss incurred after running PATE-ASQ might be smaller than the prescribed privacy budget. We refer to the privacy loss $\epsilon_{\texttt{ex post}}$, since it is determined by the output.

4.5.2 Experimental Results

The results are presented in Table 4.4, where both utility (classification accuracy on the test set) and privacy (privacy budget ϵ and privacy loss $\epsilon_{\text{ex post}}$) metrics are reported. Best results in each category are marked in bold fonts. We make a few observations of the results below.

1. Given the same privacy budget, ASQ performs substantially better than PSQ in most cases. The improvement is sometimes 10% (real-sim / $\epsilon = 0.5$). The only exception is when $\epsilon = 2.0$ on the "mushroom" dataset, in which the active learning performed substantially worse than the passive-learning counterpart in the non-private baseline as well.

2. ASQ incurs a smaller private loss $\epsilon_{\texttt{ex post}}$ than PSQ, due to possibly fewer queries being selected by the active learning algorithm than the pre-specified query budget.

3. As ϵ increases, less noise is injected by the Gaussian mechanisms, and the performance improves for both PSQ and ASQ. In the regime of small ϵ (stronger privacy), we often see a greater improvement in ASQ.

4. ASQ requires privately releasing a much smaller number of labels while maintaining comparable performances as PSQ. Although ASQ algorithms use up all labeling

Table 4.4: Utility and privacy results of different PATE models. **# Queries** shows the number of queries actually answered in experiments. **Accuracy** is reported as $\texttt{mean} \pm 1.96 \times \texttt{standard\_error}/\sqrt{30}$, i.e., 98% asymptotic confidence interval of the expected accuracy based on inverting Wald's test. All "PATE-" prefixes of methods are omitted to improve readability.

Dataset	Method	# Queries	ϵ	$\epsilon_{\texttt{ex post}}$	Accuracy
mushroom	PSQ-NP	163	$+\infty$	$+\infty$	$\mathbf{0.9773} \pm 0.0006$
	ASQ-NP	47.3 ± 0.2	$+\infty$	$+\infty$	0.9146 ± 0.0036
	PSQ	163	0.5	0.5	0.6416 ± 0.0036
	ASQ	40.1 ± 0.7	0.5	**0.4461**	$\mathbf{0.6418} \pm 0.0091$
	PSQ	163	1.0	1.0	0.7534 ± 0.0045
	ASQ	42.9 ± 0.5	1.0	**0.9267**	$\mathbf{0.7727} \pm 0.0098$
	PSQ	163	2.0	2.0	$\mathbf{0.8974} \pm 0.0027$
	ASQ	46.5 ± 0.3	2.0	**1.9410**	0.8858 ± 0.0059
a9a	PSQ-NP	977	$+\infty$	$+\infty$	$\mathbf{0.5555} \pm 0.0157$
	ASQ-NP	225.6 ± 5.0	$+\infty$	$+\infty$	0.5461 ± 0.0160
	PSQ	977	0.5	0.5	0.5040 ± 0.0034
	ASQ	293	0.5	0.5	$\mathbf{0.5212} \pm 0.0088$
	PSQ	977	1.0	1.0	0.5171 ± 0.0050
	ASQ	290.8 ± 0.8	1.0	**0.9958**	$\mathbf{0.5369} \pm 0.0103$
	PSQ	977	2.0	2.0	0.5176 ± 0.0070
	ASQ	290.3 ± 0.9	2.0	**1.9896**	$\mathbf{0.5543} \pm 0.0089$
real-sim	PSQ-NP	$1,447$	$+\infty$	$+\infty$	0.8234 ± 0.0014
	ASQ-NP	434	$+\infty$	$+\infty$	$\mathbf{0.8289} \pm 0.0008$
	PSQ	$1,447$	0.5	0.5	0.6355 ± 0.0065
	ASQ	434	0.5	0.5	$\mathbf{0.7389} \pm 0.0014$
	PSQ	$1,447$	1.0	1.0	0.7550 ± 0.0058
	ASQ	434	1.0	1.0	$\mathbf{0.8040} \pm 0.0009$
	PSQ	$1,447$	2.0	2.0	0.8025 ± 0.0037
	ASQ	434	2.0	2.0	$\mathbf{0.8231} \pm 0.0009$

budget on real-sim datasets, ASQ algorithms do not run out of them on mushroom and a9a datasets in most cases.

5. ASQ-NP does not always perform better than PSQ-NP algorithms, which meets our understanding from active learning literature. It only performs better than PSQ-NP on real-sim datasets.

4.6 Conclusions

Existing theoretical analysis shows that PATE framework consistently learns any VC-classes in the realizable setting, but not in the more general cases. We show that PATE learns any VC-classes under Tsybakov noise condition (TNC) with fast rates. When specializing to the realizable case, our results improve the best known sample complexity bound for both the public and private data. We show that PATE is incompatible with the agnostic learning setting because it is essentially trying to learn a different class of voting classifiers which could be better, worse, or comparable to the best classifier in the base-class. Lastly, we investigated the PATE framework with active learning and showed that simple Gaussian mechanism suffices for obtaining the same fast rates under TNC. In addition, our experiments on PATE-ASQ show it works as an efficient algorithm in practice.

Future work includes understanding different selections made by sparse vector technique and active learning, as well as addressing the open theoretical problem *at large* — developing ERM-oracle efficient algorithm for the private agnostic learning when a public unlabeled dataset is available.

4.7 Complete Proofs

4.7.1 Proofs of Existing Results

In this subsection, we provide the privacy analysis as well as reproving the results of [88] in our notation so that it becomes clear where the improvement is coming from.

Theorem 4.7.1 (Restatement of Theorem 4.3.2) *Algorithm 4 and 5 are both (ϵ, δ)-DP.*

The proof for Algorithm 4 follows straightforwardly from Gaussian mechanism because the number of "teachers" who predict 1 will have a global sensitivity of 1. The proof for Algorithm 5 is more delicate. It follows the arguments in the proof of Theorem 3.6 of [106] for the most part, which combines the *sparse vector technique* (SVT) [113] with the *distance to stability* approach from [114]. The only difference in the stated result here is that we used the modern CDP approach to handle the composition which provides tighter constants.

Proof: First note that the global sensitivity (Definition B.0.2) of the vote count is 1. Algorithm 4 is a straightforward adaptive composition of ℓ Gaussian mechanisms (Lemma B.0.5), which satisfies $\frac{\ell}{2\sigma^2}$-zCDP. By Lemma B.0.12, we get that the choice of σ gives us (ϵ, δ)-differential privacy.

Let us now address Algorithm 5. First note that $\widehat{\Delta}(x_j)$ as a function of the input dataset D has a global sensitivity of 2 for all x_j, thus dist_j has a global sensitivity of 1. Following the proof of Theorem 3.6 of [106], Algorithm 5 can be considered a composition of Sparse Vector Technique (SVT) (Algorithm 9), which outputs a binary vector of $\{\perp, \top\}$ indicating the failures and successes of passing the screening by SVT, and the distance-to-stability mechanism (Algorithm 10) which outputs $\{\hat{h}^{\texttt{agg}}(x_j)\}$ for all coordinates where the output is $\perp$. Check that the length of this binary vector is random

and is between T and ℓ. The number of $\top$ is smaller than T. If $\{\hat{h}^{\texttt{agg}}(x_j)\}$ is not revealed, then this would be the standard SVT, and the challenge is to add the additional outputs.

The key trick of the proof inspired from the privacy analysis (Lemma B.0.9) of the distance-to-stability is to discuss the two cases. In the first case, assume for all j such that the output is $\perp$, $\hat{h}^{\texttt{agg}}(x_j)$ remains the same over D, D', then adding $\hat{h}^{\texttt{agg}}(x_j)$ to the output obeys 0-DP; in the second case, assume that there exists some j where we output $\perp$ such that, $\hat{h}^{\texttt{agg}}(x_j)$ is different under D and D', then for all these j we know that $\text{dist}_j = 0$ for both D and D'. By the choice of λ, w, we know that the second case happens with probability at most $\delta/2$ using the tail of Laplace distribution and a union bound over all $\ell + T$ independent Laplace random variables. Note that this holds uniformly over all possible adaptive choices of the nature, since this depends only on the added noise.

Conditioning on the event that the second case does not happen, the output of the algorithm is only the binary vector of $\{\perp, \top\}$ from SVT. The SVT with cutoff T is an adaptive composition of T SVTs with cutoff$= 1$. By our choice of parameter λ, each such SVT with cutoff$= 1$ obeys pure-DP with privacy parameter $2/\lambda$, hence also satisfy CDP with parameter $2/\lambda^2$ by Proposition 1.4 of [105]. Composing over T SVTs, we get a CDP parameter of $2T/\lambda^2$. By Proposition 1.3 of [105] (Lemma B.0.12), we can convert CDP to DP. The choice of λ is chosen such that the composed mechanism obeys $(\epsilon, \delta/2)$-DP. Combining with the second case above, this establishes the (ϵ, δ)-DP of Algorithm 5. ■

Theorem 4.7.2 (Restatement of Theorem 4.3.5) *Set*

$$T = 3\Big(\mathbb{E}[\texttt{Err}(\hat{h}_1)]m + \sqrt{\frac{m\log(m/\beta)}{2}}\Big),$$

$$K = O\Big(\frac{\log(mT/\min(\delta,\beta))\sqrt{T\log(1/\delta)}}{\epsilon}\Big).$$

Let $\hat{h}^S$ be the output of Algorithm 6 that uses Algorithm 5 for privacy aggregation. With probability at least $1-\beta$ (over the randomness of the algorithm and the randomness of all data points drawn i.i.d.), we have

$$\texttt{Err}(\hat{h}^S) \leq \tilde{O}\Big(\frac{d^2 m \log(1/\delta)}{n^2\epsilon^2} + \sqrt{\frac{d}{m}}\Big)$$

for the realizable case, and

$$\texttt{Err}(\hat{h}^S) \leq 13\texttt{Err}(h^*) + \tilde{O}\Big(\frac{m^{1/3}d^{2/3}}{n^{2/3}\epsilon^{2/3}} + \sqrt{\frac{d}{m}}\Big)$$

for the agnostic case.

Proof: The analysis essentially follows the proof of Theorem 4.4.3 by replacing h^* with y. First, by Hoeffding's inequality, with probability $1-\beta$ over the teacher data points, the total number of mistakes made by each teacher classifier is at most $m\mathbb{E}[\texttt{Err}(\hat{h}_1)]+\sqrt{m\log(m/\beta)/2}$, which is B in Lemma 4.3.4. Then following Lemma 4.3.4, by choose $T = 3B = 3(m\mathbb{E}[\texttt{Err}(\hat{h}_1)] + \sqrt{m\log(m/\beta)/2})$, we ensure that the majority voting classifiers are correct and have high margin in at least $m-T$ examples.

In the realizable setting. Since $\texttt{Err}(h^*) = 0$ and by standard statistical learning theory in the realizable case (Lemma A.0.8), for each teacher classifier $\hat{h}_k$ we have

$$\texttt{Err}(\hat{h}_k) \leq 4\frac{d\log(n/K) + \log(4/\gamma)}{n/K}.$$

Substitute our choice of $K = \tilde{O}(\sqrt{T\log(1/\delta)}/\epsilon)$ as in Lemma 4.3.3 we get that w.h.p.

$$\texttt{Err}(\hat{h}_k) \leq \tilde{O}\Big(\frac{d\sqrt{T\log(1/\delta)}}{n\epsilon}\Big).$$

Plug in the bound into our choice of $T = 3(m\mathbb{E}[\texttt{Err}(\hat{h}_1)] + \sqrt{m\log(m/\beta)/2})$, we get

$$T \leq \tilde{O}\Big(\frac{dm\sqrt{T\log(1/\delta)}}{n\epsilon} + \sqrt{\frac{m\log(m/\beta)}{2}}\Big).$$

By solving the quadratic inequality, we get that T obeys

$$T \leq \tilde{O}\Big(\frac{d^2m^2\log(1/\delta)}{n^2\epsilon^2} + \sqrt{m}\Big).$$

Recall that this choice of K and T ensures that Algorithm 5 will have at most T unstable queries during the m rounds, which implies that with high probability, the privately released pseudo-labels to those "stable" queries are the same as the corresponding true labels.

Now the next technical subtlety is to deal with the dependence in the student learning problem created by the pseudo-labels via a reduction to an ERM learner. By the standard Hoeffding-style uniform convergence bound (Lemma A.0.7),

$$\begin{aligned}
\texttt{Err}(\hat{h}^S) &\leq \widehat{\texttt{Err}}(\hat{h}^S) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\
&\leq \widehat{\texttt{Err}}(\hat{h}^{\texttt{priv}}) + \widehat{\text{Dis}}(\hat{h}^{\texttt{priv}}, \hat{h}^S) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\
&\leq 2\widehat{\texttt{Err}}(\hat{h}^{\texttt{priv}}) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\
&\leq \frac{2T}{m} + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\
&= \tilde{O}\Big(\frac{d^2m\log(1/\delta)}{n^2\epsilon^2} + \sqrt{\frac{d}{m}}\Big). \qquad (4.5)
\end{aligned}$$

where we applied the triangular inequality in the second line, used that $\hat{h}^S$ is the minimizer of $\widehat{\text{Dis}}(\hat{h}^{\texttt{priv}}, \cdot)$ in the third line, and then combined Lemma 4.3.3 and Lemma 4.3.4 to show that under the appropriate choice of T and K with high probability, $\hat{h}^{\texttt{priv}}(x_j)$

correctly returns y_j except for up to T example. Finally, the choice of T is substituted.

In agnostic setting. By Lemma A.0.8, with high probability, for all teacher classifier $\hat{h}_k$ for $k = 1, ..., K$, we have

$$\texttt{Err}(\hat{h}_k) - \texttt{Err}(h^*) \leq \tilde{O}\Big(\sqrt{\frac{d\log(n/K) + \log(4/\gamma)}{n/K}}\Big).$$

Substitute the choice of $K = \tilde{O}(\sqrt{T\log(1/\delta)}/\epsilon)$ from Lemma 4.3.3, we get

$$\texttt{Err}(\hat{h}_k) \leq \texttt{Err}(h^*) + \tilde{O}\Big(\frac{d^{1/2}T^{1/4}}{n^{1/2}\epsilon^{1/2}}\Big).$$

Plug in the above bound into our choice $T = 3(m\mathbb{E}[\texttt{Err}(\hat{h}_1)] + \sqrt{m\log(m/\beta)/2})$, we get that

$$T \leq 3m\texttt{Err}(h^*) + \tilde{O}(\sqrt{m}) + \tilde{O}\Big(\frac{md^{1/2}T^{1/4}}{n^{1/2}\epsilon^{1/2}}\Big). \tag{4.6}$$

Further, we can write

$$\begin{aligned} T &\leq 2(3m\texttt{Err}(h^*) + \tilde{O}(\sqrt{m})) \cdot \mathbb{1}\Big(\tilde{O}\Big(\frac{md^{1/2}T^{1/4}}{n^{1/2}\epsilon^{1/2}}\Big) \leq \frac{T}{2}\Big) \\ &\quad + \Big(2\tilde{O}\Big(\frac{md^{1/2}}{n^{1/2}\epsilon^{1/2}}\Big)\Big)^{4/3} \cdot \mathbb{1}\Big(\tilde{O}\Big(\frac{md^{1/2}T^{1/4}}{n^{1/2}\epsilon^{1/2}}\Big) > \frac{T}{2}\Big) \\ &\leq 6m\texttt{Err}(h^*) + \tilde{O}(\sqrt{m}) + \tilde{O}\Big(\frac{m^{4/3}d^{2/3}}{n^{2/3}\epsilon^{2/3}}\Big), \end{aligned} \tag{4.7}$$

where the first line talks about two cases of Inequality (4.6): (1) $T/2 \leq T - \tilde{O}(\frac{md^{1/2}T^{1/4}}{n^{1/2}\epsilon^{1/2}}) \leq 3m\texttt{Err}(h^*) + \tilde{O}(\sqrt{m})$ if $\tilde{O}(\frac{md^{1/2}T^{1/4}}{n^{1/2}\epsilon^{1/2}}) \leq T/2$, and (2) $T^{3/4} \leq 2\tilde{O}(\frac{md^{1/2}T^{1/4}}{n^{1/2}\epsilon^{1/2}})$ if $\tilde{O}(\frac{md^{1/2}T^{1/4}}{n^{1/2}\epsilon^{1/2}}) > T/2$; The second line is due to the indicator function is always ≤ 1.

Similar to the realizable case, now we apply a reduction to ERM. By the Hoeffding's

style uniform convergence bound (implied by Lemma A.0.7)

$$\begin{aligned}
\texttt{Err}(\hat{h}^S) &\leq \widehat{\texttt{Err}}(\hat{h}^S) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\
&\leq \widehat{\texttt{Err}}(\hat{h}^{\texttt{priv}}) + \widehat{\text{Dis}}(\hat{h}^{\texttt{priv}}, \hat{h}^S) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\
&\leq \widehat{\texttt{Err}}(\hat{h}^{\texttt{priv}}) + \widehat{\text{Dis}}(\hat{h}^{\texttt{priv}}, \hat{h}_1) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\
&\leq 2\widehat{\texttt{Err}}(\hat{h}^{\texttt{priv}}) + \widehat{\texttt{Err}}(\hat{h}_1) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\
&\leq \frac{2T}{m} + \texttt{Err}(h^*) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\
&\leq 13\texttt{Err}(h^*) + \tilde{O}\Big(\frac{m^{1/3}d^{2/3}}{n^{2/3}\epsilon^{2/3}} + \sqrt{\frac{d}{m}}\Big).
\end{aligned}$$

where the second and fourth lines use the triangular inequality of $0-1$ error, the third line uses the fact that $\hat{h}^S$ is the empirical risk minimizer of the student learning problem with labels $\hat{h}^{\texttt{priv}}$ and the fact that $h_1 \in \mathcal{H}$. The second last line follows from the fact that in those stable queries $\hat{h}^{\texttt{priv}}(x_j)$ outputs y_j, and a standard agnostic learning bound. Finally, in the last line, we obtain the stated result by substituting the upper bound of T from (4.7). ■

The results stated in Table 4.1 are obtained by minimizing the bound by choosing a random subset of data points to privately release labels.

4.7.2 Learning bound for PATE with Gaussian Mechanism

In this subsection, we provide a theoretical analysis of the version of PATE from [85, 86] that uses Gaussian mechanism to release the aggregated teacher labels. We will focus on the setting assuming τ-TNC. Though this result is not our main contribution, we note that standard PATE is a practical algorithm and this is the first learning-theoretic guarantees of PATE.

Theorem 4.7.3 (Utility guarantee of Algorithm 4) *Assume the data distribution $\mathcal{D}$ and the hypothesis class $\mathcal{H}$ obey the Tsybakov noise condition with parameter τ, then with probability at least $1-\gamma$, there exists universal constant C such that the output $\hat{h}_S$ of Algorithm 4 with parameter K satisfying*

$$K = \frac{6\sqrt{\log(2n)}(\sqrt{m\log(1/\delta)} + \sqrt{m\log(1/\delta) + \epsilon m})}{\epsilon}$$

obeys that

$$\texttt{Err}(\hat{h}^S) - \texttt{Err}(h^*) \leq \tilde{O}\Big(\frac{d}{m} + \Big(\frac{d\sqrt{m}}{n\epsilon}\Big)^{\frac{\tau}{2-\tau}}\Big).$$

Specifically, in the realizable setting, then it follows that

$$\texttt{Err}(\hat{h}^S) - \texttt{Err}(h^*) \leq \tilde{O}\Big(\frac{d}{m} + \frac{d\sqrt{m}}{n\epsilon}\Big).$$

Proof: By the triangular inequality of the $0-1$ error,

$$\begin{aligned}
\texttt{Err}(\hat{h}^S) - \texttt{Err}(h^*) &\leq \mathrm{Dis}(\hat{h}^S, h^*) \\
&\leq \mathrm{Dis}(\hat{h}^S, \tilde{h}^{\texttt{priv}}) + \mathrm{Dis}(\tilde{h}^{\texttt{priv}}, h^*) \\
&\leq 2\mathrm{Dis}(\tilde{h}^{\texttt{priv}}, h^*) + 2\sqrt{\frac{(d+\log(4/\gamma))\mathrm{Dis}(\tilde{h}^{\texttt{priv}}, h^*)}{m}} + \frac{4(d+\log(4/\gamma))}{m} \\
&\leq 4\mathrm{Dis}(\tilde{h}^{\texttt{priv}}, h^*) + \tilde{O}\Big(\frac{d}{m}\Big) \qquad (4.8)
\end{aligned}$$

where the third line follows from the learning bound (Lemma A.0.8) with $\tilde{h}^{\texttt{priv}}$ being the labeling function for the student dataset. The last line is due to $a + 2\sqrt{ab} + b \leq 2a + 2b$ for non-negative a, b.

The remaining problem would be finding the upper bound of $\mathrm{Dis}(\tilde{h}^{\texttt{priv}}, h^*)$. First by

Lemma 4.4.6, with probability at least $1-\gamma/2$, $\forall k \in [K]$ we have

$$\mathrm{Dis}(\hat{h}_k, h^*) \lesssim \eta^{\frac{2}{2-\tau}} \Big(\frac{dK\log(n/d) + \log(2K/\gamma)}{n} \Big)^{\frac{\tau}{2-\tau}}.$$

Next, conditioning on the teachers, $\tilde{h}^{\mathtt{priv}}$ is independent for each input and well-defined for all input. Let $Z \sim \mathcal{N}(0, \sigma^2)$. By Gaussian-tail bound and Markov's inequality,

$$\begin{aligned}
&\mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h^*) \\
&\le \mathbb{P}\Big[|Z| \le \sigma\sqrt{2\log\big(\frac{2}{\beta}\big)}\Big] \mathbb{P}\Big[\sum_{k=1}^{K} \mathbb{1}(\hat{h}_k(x) \neq h^*(x)) \ge \frac{K}{2} - |Z| \Big| |Z| \le \sigma\sqrt{2\log\big(\frac{2}{\beta}\big)}\Big] \\
&\quad + \mathbb{P}\Big[|Z| > \sigma\sqrt{2\log\big(\frac{2}{\beta}\big)}\Big] \\
&\le \frac{1}{K/2 - \sigma\sqrt{2\log(2/\beta)}} \sum_{k=1}^{K} \mathbb{E}[\mathbb{1}(\hat{h}_k(x) \neq h^*(x))] + \beta \\
&\le \frac{3}{K} \sum_{k=1}^{K} \mathrm{Dis}(\hat{h}_k, h^*) + \frac{1}{n} \\
&\lesssim \eta^{\frac{2}{2-\tau}} \Big(\frac{dK\log(n/d) + \log(2K/\gamma)}{n} \Big)^{\frac{\tau}{2-\tau}}.
\end{aligned}$$

In the last line, we choose $\beta = 1/n$ and applied the assumption that $K \ge 6\sigma\sqrt{2\log(2n)}$.

Note that our choice of σ satisfies that

$$\sqrt{\frac{2m\log(1/\delta)}{\sigma^2}} + \frac{m}{2\sigma^2} = \epsilon.$$

Solve the equation and we find that

$$\sigma = \frac{\sqrt{2m\log(1/\delta)} + \sqrt{2m\log(1/\delta) + 2\epsilon m}}{2\epsilon}.$$

Therefore, the choice of K is

$$K = \frac{6\sqrt{\log(2n)}(\sqrt{m\log(1/\delta)} + \sqrt{m\log(1/\delta) + \epsilon m})}{\epsilon} = \tilde{O}\Big(\frac{\sqrt{m}}{\epsilon}\Big),$$

where ϵ is assumed to be small. Put everything together, and the excess risk bound is

$$\texttt{Err}(\hat{h}^S) - \texttt{Err}(h^*) \leq \tilde{O}\left(\frac{d}{m} + \left(\frac{d\sqrt{m}}{n\epsilon}\right)^{\frac{\tau}{2-\tau}}\right).$$

■

Remark 4.7.4 *When m is sufficient large ($\frac{d}{m} < \left(\frac{d\sqrt{m}}{n\epsilon}\right)^{\frac{\tau}{2-\tau}}$), it suffices to use a subset of randomly chosen data points to optimize the bound and we obtain an excess risk bound of $\tilde{O}\Big(\big(\frac{d^{3/2}}{n\epsilon}\big)^{\frac{2\tau}{4-\tau}}\Big)$. When $\tau = 1$, this yields the $\frac{d}{(n\epsilon)^{2/3}}$ rate that matches [88]'s analysis of SVT-based PATE. To avoid any confusions, Gaussian mechanism-based PATE is still theoretically inferior comparing to SVT-based PATE as we established in Theorem 4.4.3.*

4.7.3 Deferred Proofs of Results in Main Sections

In this subsection, we present full proofs of our results shown in main sections.

Proposition 4.7.5 (Restatement of Proposition 4.4.13) *Assume the learning problem with n/K i.i.d. data points satisfies (ν, ξ)-approximate high-margin condition. Let Algorithm 5 be instantiated with parameters*

$$T \geq \nu m + \sqrt{2\nu m \log\left(\frac{3}{\gamma}\right)} + \frac{2}{3}\log\left(\frac{3}{\gamma}\right)$$

$$K \geq \max\left\{\frac{2\log(3m/\gamma)}{\xi^2}, \frac{3\lambda\big(\log(4m/\delta) + \log(3m/\gamma)\big)}{\xi}\right\},$$ [7]

[7] $\lambda = (\sqrt{2T(\epsilon + \log(2/\delta))} + \sqrt{2T\log(2/\delta)})/\epsilon$ according to Algorithm 5.

then with high probability (over the randomness of the n i.i.d. samples of the private dataset, m i.i.d. samples of the public dataset, and that of the randomized algorithm), Algorithm 5 finishes all m rounds and the output is the same as $h_\infty^{\mathtt{agg}}(x_i)$ for all but T of the $i \in [m]$.

Proof: By the Bernstein's inequality, with probability $\geq 1 - \gamma_2$ over the i.i.d. samples of the public data, the number of queries $j \in [m]$ with $\Delta_{n/K}(x_j) \leq \xi$ is smaller than $\nu m + \sqrt{2\nu m \log(1/\gamma_2)} + \frac{2}{3}\log(1/\gamma_2)$. T is an upper bound of the above quantity if we choose $\gamma_2 = \gamma/3$.

Conditioning on the above event, by Hoeffding's inequality and a union bound, with probability $\geq 1-\gamma_3$ over the i.i.d. samples of the private data (hence the K i.i.d. teacher classifiers), for all $m-T$ queries with $\Delta_{n/K}(x_i)$ larger than ξ, the realized margin (defined in (4.1)) obeys that

$$\begin{aligned}\widehat{\Delta}(x_j) &\geq \mathbb{E}[\widehat{\Delta}(x_j)|x_j] - \sqrt{2K\log\left(\frac{m}{\gamma_3}\right)} \\ &= 2K\Delta_{n/K}(x_i) - \sqrt{2K\log\left(\frac{m}{\gamma_3}\right)} \\ &\geq 2K\xi - \sqrt{2K\log\left(\frac{m}{\gamma_3}\right)}.\end{aligned}$$

It remains to check that under our choice of T, K, $\widehat{\mathrm{dist}}_j > \hat{w}$ for all $j \in [m]$ except the (up to) T exceptions.

By the tail of Laplace distribution and a union bound, with probability $\geq 1 - \gamma_1$, all m Laplace random variables that perturb the distance to stability $\widehat{\mathrm{dist}}_j$ in Algorithm 10 is larger than $-2\lambda\log((m+T)/(2\gamma_1))$ and all T Laplace random variables that perturb the threshold w is smaller than $\lambda\log((m+T)/(2\gamma_1))$, where λ is chosen according to Algorithm 5. We simplify the above bound by using $T < m$.

It suffices that K is chosen such that

$$2K\xi - \sqrt{2K\log\Big(\frac{m}{\gamma_3}\Big)} - 2\lambda\log\Big(\frac{m}{\gamma_1}\Big) > w + \lambda\log\Big(\frac{m}{\gamma_1}\Big).$$

Substitute Algorithm 5's choice $w = 3\lambda\log(2(m+T)/\delta) \le 3\lambda\log(4m/\delta)$. Assume $K \ge 2\log(m/\gamma_3)/\xi^2$, we have $2K\xi - \sqrt{2K\log(m/\gamma_3)} \ge K\xi$, thus it suffices that further $K\xi > 3\lambda\big(\log(4m/\delta) + \log(m/\gamma_1)\big)$.

The proof is complete by taking $\gamma_2 = \gamma_3 = \gamma/3$ and take union bound over all high probability events described above. ■

Theorem 4.7.6 (Restatement of Theorem 4.4.14) *Assume the learning problem with n/K i.i.d. data points satisfies (ν,ξ)-approximate high-margin condition and let K, T be chosen according to Proposition 4.4.13, furthermore assume that the privacy parameter of choice $\epsilon \le \log(2/\delta)$, then the output classifier $\hat{h}^S$ of Algorithm 6 in the agnostic setting satisfies that with probability $\ge 1-2\gamma$,*

$$\mathtt{Err}(\hat{h}^S) - \mathtt{Err}(h_\infty^{\mathtt{agg}}) \le \min_{h\in\mathcal{H}} \mathrm{Dis}(h, h_\infty^{\mathtt{agg}}) + \frac{2T}{m} + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \le \min_{h\in\mathcal{H}} \mathrm{Dis}(h, h_\infty^{\mathtt{agg}}) + 2\nu + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big).$$

Proof: We follow a similar argument as in the proof of Theorem 4.4.3, but replace h^* with $h_\infty^{\mathtt{agg}}$. Define $\tilde{h} = \mathrm{argmin}_{h\in\mathcal{H}} \widehat{\mathrm{Dis}}(h, h_\infty^{\mathtt{agg}})$. By the triangular inequality of the $0-1$ error and Lemma A.0.7 in Appendix A,

$$\mathtt{Err}(\hat{h}^S) - \mathtt{Err}(h_\infty^{\mathtt{agg}}) \le \mathrm{Dis}(\hat{h}^S, h_\infty^{\mathtt{agg}}) \le \widehat{\mathrm{Dis}}(\hat{h}^S, h_\infty^{\mathtt{agg}}) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big). \tag{4.9}$$

By the triangular inequality, we have $\widehat{\mathrm{Dis}}(\hat{h}^S, h_\infty^{\mathtt{agg}}) \le \widehat{\mathrm{Dis}}(\hat{h}^S, \hat{h}^{\mathtt{priv}}) + \widehat{\mathrm{Dis}}(\hat{h}^{\mathtt{priv}}, h_\infty^{\mathtt{agg}})$,

therefore,

$$\begin{aligned}(4.9) &\le \widehat{\mathrm{Dis}}(\hat{h}^S, \hat{h}^{\mathtt{priv}}) + \widehat{\mathrm{Dis}}(\hat{h}^{\mathtt{priv}}, h_\infty^{\mathtt{agg}}) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\ &\le \widehat{\mathrm{Dis}}(\tilde{h}, \hat{h}^{\mathtt{priv}}) + \widehat{\mathrm{Dis}}(\hat{h}^{\mathtt{priv}}, h_\infty^{\mathtt{agg}}) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\ &\le \widehat{\mathrm{Dis}}(\tilde{h}, h_\infty^{\mathtt{agg}}) + 2\widehat{\mathrm{Dis}}(\hat{h}^{\mathtt{priv}}, h_\infty^{\mathtt{agg}}) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big) \\ &\le \min_{h\in\mathcal{H}} \mathrm{Dis}(h, h_\infty^{\mathtt{agg}}) + 2\widehat{\mathrm{Dis}}(\hat{h}^{\mathtt{priv}}, h_\infty^{\mathtt{agg}}) + \tilde{O}\Big(\sqrt{\frac{d}{m}}\Big).\end{aligned}$$

In the second line, we applied the fact that $\hat{h}^S = \mathrm{argmin}_{h\in\mathcal{H}} \widehat{\mathrm{Dis}}(h, \hat{h}^{\mathtt{priv}})$; in the third line, we applied triangular inequality again and the last line is true because $\tilde{h} = \mathrm{argmin}_{h\in\mathcal{H}} \widehat{\mathrm{Dis}}(h, h_\infty^{\mathtt{agg}})$.

Recall that T is the unstable cutoff in Algorithm 6. The proof completes by invoking Proposition 4.4.13 which implies that $\widehat{\mathrm{Dis}}(\hat{h}^{\mathtt{priv}}, h_\infty^{\mathtt{agg}}) \le T/m$ with high probability.

■

Theorem 4.7.7 (Restatement of Theorem 4.4.15) *Under the same assumption of Theorem 4.4.14, suppose we train an ensemble classifier within the voting hypothesis space* $\mathrm{Vote}_K(\mathcal{H})$ *in the student domain, then the output classifier* $\hat{h}^S$ *of Algorithm 6 in the agnostic setting satisfies that with probability* $\ge 1 - 2\gamma$,

$$\mathtt{Err}(\hat{h}^S) - \mathtt{Err}(h_\infty^{\mathtt{agg}}) \le \frac{4T}{m} + \frac{5(Kd + \log(4/\gamma))}{m} = \tilde{O}\Big(\nu + \frac{\log(4/\gamma)}{m} + \frac{d\sqrt{\nu}}{\xi\sqrt{m}}\Big).$$

Proof: Define $\hat{h}^S = \mathrm{argmin}_{h\in\mathrm{Vote}_K(\mathcal{H})} \widehat{\mathrm{Dis}}(h, \hat{h}^{\mathtt{priv}})$ and $\tilde{h} = \mathrm{argmin}_{h\in\mathrm{Vote}_K(\mathcal{H})} \widehat{\mathrm{Dis}}(h, h_\infty^{\mathtt{agg}})$.

By the triangular inequality of the $0-1$ error,

$$\begin{aligned}\texttt{Err}(\hat{h}^S) - \texttt{Err}(h_\infty^{\texttt{agg}}) &\leq \mathrm{Dis}(\hat{h}^S, h_\infty^{\texttt{agg}}) \\ &\leq \widehat{\mathrm{Dis}}(\hat{h}^S, h_\infty^{\texttt{agg}}) + 2\sqrt{\frac{(Kd+\log(4/\gamma))\widehat{\mathrm{Dis}}(\hat{h}^S, h_\infty^{\texttt{agg}})}{m}} + \frac{4(Kd+\log(4/\gamma))}{m} \\ &\leq 2\widehat{\mathrm{Dis}}(\hat{h}^S, h_\infty^{\texttt{agg}}) + \frac{5(Kd+\log(4/\gamma))}{m}, \qquad (4.10)\end{aligned}$$

where the second line follows from the first statement of Lemma A.0.7 in Appendix A with $z = h_\infty^{\texttt{agg}}(x)$ and the third line is due to $a + 2\sqrt{ab} + b \leq 2a + 2b$ for non-negative a, b.

By the triangular inequality, we have $\widehat{\mathrm{Dis}}(\hat{h}^S, h_\infty^{\texttt{agg}}) \leq \widehat{\mathrm{Dis}}(\hat{h}^S, \hat{h}^{\texttt{priv}}) + \widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h_\infty^{\texttt{agg}})$, therefore,

$$\begin{aligned}(4.10) &\leq 2\widehat{\mathrm{Dis}}(\hat{h}^S, \hat{h}^{\texttt{priv}}) + 2\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h_\infty^{\texttt{agg}}) + \frac{5(Kd+\log(4/\gamma))}{m} \\ &\leq 2\widehat{\mathrm{Dis}}(\tilde{h}, \hat{h}^{\texttt{priv}}) + 2\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h_\infty^{\texttt{agg}}) + \frac{5(Kd+\log(4/\gamma))}{m} \\ &\leq 2\widehat{\mathrm{Dis}}(\tilde{h}, h_\infty^{\texttt{agg}}) + 4\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h_\infty^{\texttt{agg}}) + \frac{5(Kd+\log(4/\gamma))}{m} \\ &\leq 4\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h_\infty^{\texttt{agg}}) + \frac{5(Kd+\log(4/\gamma))}{m}.\end{aligned}$$

In the second line, we applied the fact that $\hat{h}^S = \mathrm{argmin}_{h \in \mathrm{Vote}_K(\mathcal{H})} \widehat{\mathrm{Dis}}(h, \hat{h}^{\texttt{priv}})$; in the third line, we applied triangular inequality again and the last line is true because $\widehat{\mathrm{Dis}}(\tilde{h}, h_\infty^{\texttt{agg}}) = 0$ since $\tilde{h}$ is the minimizer and that $h_\infty^{\texttt{agg}} \in \mathrm{Vote}_K(\mathcal{H})$.

Recall that T is the unstable cutoff in Algorithm 6. The proof completes by using that $\widehat{\mathrm{Dis}}(\hat{h}^{\texttt{priv}}, h_\infty^{\texttt{agg}}) \leq T/m$ with probability $1-\gamma$ according to Proposition 4.4.13 and substitute the choices of T and K accordingly. ■

Lemma 4.7.8 *If the disagreement-based agnostic active learning algorithm is given a stream of m unlabeled data points, then with probability at least $1-\gamma$, the algorithm*

returns a hypothesis h obeying that,

$$\texttt{Err}(h) - \texttt{Err}(h^*) \lesssim \frac{d\log(\theta(d/m)) + \log(1/\gamma)}{m} + \sqrt{\frac{\texttt{Err}(h^*)(d\log(\theta(\texttt{Err}(h^*)) + \log(1/\gamma))}{m}}.$$

Proof: From Lemma 3.1 of [93], we learn that for any hypothesis h survive in version space V must satisfy

$$\texttt{Err}(h) - \texttt{Err}(h^*) \leq 2U(m, \gamma).$$

Then by the definition of $U(m, \gamma)$ shown in Algorithm 7, we have

$$\texttt{Err}(h) - \texttt{Err}(h^*) \lesssim \frac{d\log(\theta(d/m)) + \log(1/\gamma)}{m} + \sqrt{\frac{\texttt{Err}(h^*)(d\log(\theta(\texttt{Err}(h^*)) + \log(1/\gamma))}{m}}.$$

■

Theorem 4.7.9 (Restatement of Theorem 4.4.17) *With probability at least $1-\gamma$, there exists universal constants C_1, C_2 such that for all*

$$\alpha \geq C_1 \max\left\{\eta^{\frac{2}{2-\tau}}\left(\frac{dK\log(n/d) + \log(2K/\gamma)}{n}\right)^{\frac{\tau}{2-\tau}}, \frac{d\log((m+n)/d) + \log(2/\gamma)}{m}\right\},$$

the output $\hat{h}^S$ of Algorithm 8 with parameter ℓ, K satisfying

$$\ell = C_2\theta(\alpha)\Big(1 + \log\big(\frac{1}{\alpha}\big)\Big)\left(d\log(\theta(\alpha)) + \log\left(\frac{\log(1/\alpha)}{\gamma/2}\right)\right)$$

$$K = \frac{6\sqrt{\log(2n)}(\sqrt{\ell\log(1/\delta)} + \sqrt{\ell\log(1/\delta) + \epsilon\ell})}{\epsilon}$$

obeys that

$$\texttt{Err}(\hat{h}^S) - \texttt{Err}(h^*) \leq \alpha.$$

Specifically, when we choose

$$\alpha = C_1 \max\left\{\eta^{\frac{2}{2-\tau}}\Big(\frac{dK\log(n/d)+\log(2K/\gamma)}{n}\Big)^{\frac{\tau}{2-\tau}}, \frac{d\log((m+n)/d)+\log(2/\gamma)}{m}\right\},$$

and also $\epsilon \leq \log(1/\delta)$*, then it follows that*

$$\texttt{Err}(\hat{h}^S) - \texttt{Err}(h^*) = \tilde{O}\bigg(\max\Big\{\big(\frac{d^{1.5}\sqrt{\theta(\alpha)\log(1/\delta)}}{n\epsilon}\big)^{\frac{\tau}{2-\tau}}, \frac{d}{m}\Big\}\bigg),$$

where $\tilde{O}$ *hides logarithmic factors in* $m, n, 1/\gamma$*.*

Proof: **Step 1: Teachers are good.** By Lemma 4.4.6, with probability at least $1-\gamma/2$, $\forall k \in [K]$ we have

$$\mathrm{Dis}(\hat{h}_k, h^*) \lesssim \eta^{\frac{2}{2-\tau}}\Big(\frac{dK\log(n/d)+\log(2K/\gamma)}{n}\Big)^{\frac{\tau}{2-\tau}}.$$

Step 2: PATE is just as good. Let $\tilde{h}^{\texttt{priv}}$ be a randomized classifier from Line 4 of Algorithm 4. Conditioning on the teachers, this classifier is independent for each input and well-defined for all input. Note that $\hat{h}^{\texttt{priv}}$ that uses Algorithm 5 do not have these

properties. Let $Z \sim \mathcal{N}(0, \sigma^2)$. By Gaussian-tail bound and Markov's inequality,

$$
\begin{aligned}
&\mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h^*) \\
&\le \mathbb{P}\Big[|Z| \le \sigma\sqrt{2\log\big(\frac{2}{\beta}\big)}\Big]\mathbb{P}\Big[\sum_{k=1}^{K} \mathbb{1}(\hat{h}_k(x) \neq h^*(x)) \ge \frac{K}{2} - |Z| \Big| |Z| \le \sigma\sqrt{2\log\big(\frac{2}{\beta}\big)}\Big] \\
&\quad + \mathbb{P}\Big[|Z| > \sigma\sqrt{2\log\big(\frac{2}{\beta}\big)}\Big] \\
&\le \frac{1}{K/2 - \sigma\sqrt{2\log(2/\beta)}} \sum_{k=1}^{K} \mathbb{E}[\mathbb{1}(\hat{h}_k(x) \neq h^*(x))] + \beta \\
&\le \frac{3}{K}\sum_{k=1}^{K} \mathrm{Dis}(\hat{h}_k, h^*) + \frac{1}{n} \\
&\lesssim \eta^{\frac{2}{2-\tau}} \Big(\frac{dK\log(n/d) + \log(2K/\gamma)}{n}\Big)^{\frac{\tau}{2-\tau}}.
\end{aligned}
$$

In the last line, we choose $\beta = 1/n$ and applied the assumption that $K \ge 6\sigma\sqrt{2\log(2n)}$.

Step 3: Oracle reduction to active learning bounds. Note that $\tilde{h}^{\mathtt{priv}}$ is the labeling function in the student learning problem. So the above implies that the student learning problem is close to realizable:

$$
\min_{h\in\mathcal{H}} \mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h) \le \mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h^*) \lesssim \eta^{\frac{2}{2-\tau}} \Big(\frac{dK\log(n/d) + \log(2K/\gamma)}{n}\Big)^{\frac{\tau}{2-\tau}}.
$$

By the above, and the agnostic active learning bounds in Lemma A.0.10, to achieve an excess risk bound of $\alpha \ge \mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h^*) := \mathtt{Err}^*$ in the student learning problem with probability at least $1 - \gamma/2$, with unbounded m, it suffices to choose ℓ to be

$$
\begin{aligned}
&C\theta(\mathtt{Err}^* + \alpha)\Big(\frac{(\mathtt{Err}^*)^2}{\alpha^2} + \log\Big(\frac{1}{\alpha}\Big)\Big)\Big(d\log(\theta(\mathtt{Err}^* + \alpha)) + \log\Big(\frac{\log(1/\alpha)}{\gamma}\Big)\Big) \\
&\le C\theta(\alpha)(1 + \log(1/\alpha))\Big(d\log(\theta(\alpha)) + \log\Big(\frac{\log(1/\alpha)}{\gamma}\Big)\Big).
\end{aligned}
$$

This implies an error bound of

$$\begin{aligned}\mathrm{Dis}(\hat{h}_S, \tilde{h}^{\mathtt{priv}}) &\leq \min_{h\in\mathcal{H}} \mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h) + \alpha \\ &\leq \mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h^*) + \alpha \leq 2\alpha.\end{aligned}$$

When m is small, we might not have enough data points to obtain $\alpha = O(\mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h^*))$ in this case the error is dominated by our bounds in Lemma 4.7.8, which says that we can take

$$\alpha = C \max\left\{\mathtt{Err}^*, \frac{d\log(m/d) + \log(2/\gamma)}{m}\right\}.$$

Step 4 Putting everything together.

$$\begin{aligned}\mathtt{Err}(\hat{h}^S) - \mathtt{Err}(h^*) &\leq \mathrm{Dis}(\hat{h}^S, \tilde{h}^{\mathtt{priv}}) + \mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h^*) \\ &\lesssim \mathrm{Dis}(\tilde{h}^{\mathtt{priv}}, h^*) + \alpha \\ &\lesssim \eta^{\frac{2}{2-\tau}} \Big(\frac{dK\log(n/d) + \log(2K/\gamma)}{n}\Big)^{\frac{\tau}{2-\tau}} + \alpha.\end{aligned}$$

The proof is complete by substituting our choice of $K = 6\sigma\sqrt{2\log(2n)}$, and furthermore by the standard privacy calibration of the Gaussian mechanism, our choice of σ satisfies that

$$\sqrt{\frac{2\ell\log(1/\delta)}{\sigma^2}} + \frac{\ell}{2\sigma^2} = \epsilon.$$

following the specification of Algorithm 4. Solve the equation and we find that

$$\sigma = \frac{\sqrt{2\ell\log(1/\delta)} + \sqrt{2\ell\log(1/\delta) + 2\epsilon\ell}}{2\epsilon},$$

where ϵ is assumed to be small. ■

Chapter 5

Active Sample Selection for Video Semantic Segmentation

Accurate per-pixel semantic class annotations of the entire video are crucial for designing and evaluating video semantic segmentation algorithms. However, the annotations are usually limited to a small subset of the video frames due to the high annotation cost and limited budget in practice. In this chapter, we propose a novel human-in-the-loop framework called HVSA to generate semantic segmentation annotations for the entire video using only a small annotation budget. Our method alternates between active sample selection and test-time fine-tuning algorithms until annotation quality is satisfied. In particular, the active sample selection algorithm picks the most important samples to get manual annotations, where the sample can be a video frame, a rectangle, or even a super-pixel. Further, the test-time fine-tuning algorithm propagates the manual annotations of selected samples to the entire video. Real-world experiments show that our method generates highly accurate and consistent semantic segmentation annotations while simultaneously enjoys significantly small annotation cost.

5.1 Introduction

Video-level segmentation annotations are important in multiple applications such as autonomous driving [115], flight [116], and augmented reality [117]. They also facilitate model training in other tasks like video deblurring/dehazing [118, 119], action recognition [120], and 3D reconstruction [121]. However, manually annotating per-pixel semantic segmentation labels for the entire video is usually expensive [122]. Therefore, a typical method is to only sample a subset of video frames to get human annotations [122, 123]. And then given sparsely annotated frames, the method applies Label Propagation (LP) to populate annotations on selected frames to all frames to get dense annotations [124, 125, 126]. Unfortunately, these annotate-once-then-propagate methods do not utilize annotation budget efficiently.

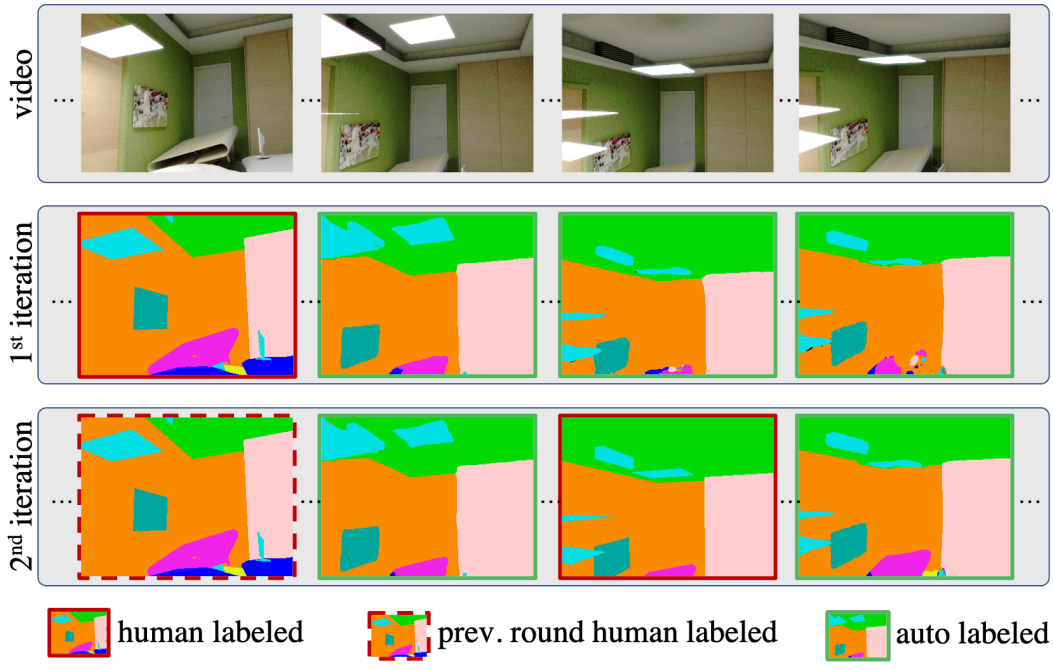


Figure 5.1: Performance of HVSA after 2 iterations. The method actively selects the most important samples to get human annotations in each iteration, then propagates the annotations to the entire video by jointly considering spatial-temporal consistency and semantic information of the video. Thus less human effort is required to obtain the high-quality pixel-level segmentation.

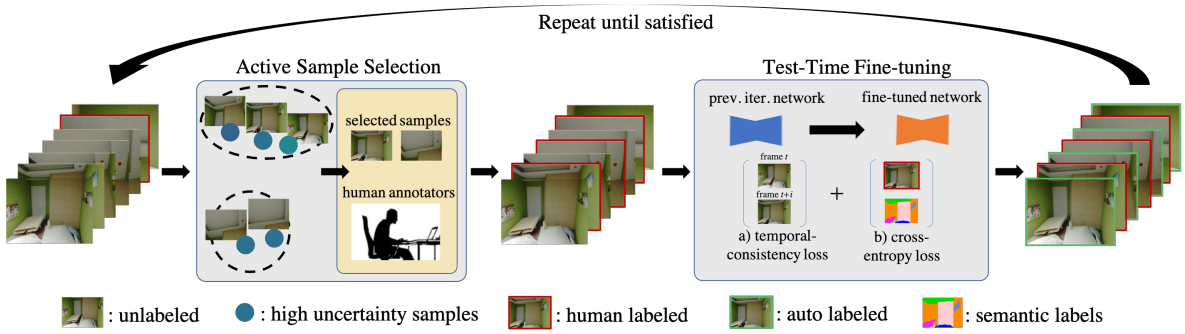


Figure 5.2: Overview of the HVSA framework. *Active sample selection* searches for uncertain and diverse samples from the input video. *Test-time fine-tuning* fine-tunes the image-based semantic segmentation network from the previous iteration by minimizing two complementary losses (a & b). The whole process repeats until high quality semantic segmentation is satisfied.

To annotate the entire video with semantic segmentation labels at a low cost, we propose the Human-in-the-loop Video Semantic segmentation Auto-annotation (HVSA) framework. Unlike most work that annotates sampled frames only *once*, our HVSA framework works *iteratively*, keeping collecting annotations and updating segmentation models at the same time unitl high quality of segmentation is satisfied. See Figure 5.2. In each iteration of HVSA, samples are *actively* selected to be manually annotated and then a video-specific network is fine-tuned based on the accumulated manual annotations. The updated outputs of the network can be used in the next iteration to decide which sample to select for human annotations. Finally, the well fine-tuned network is used to generate segmentation annotations for the entire video. To the best of our knowledge, HVSA is the first human-in-the-loop framework that applies active sample selection for efficient video semantic segmentation auto-annotation. See Figure 5.1.

To select video frames for annotation, most existing work only uses naive strategies, e.g., the first few frames, uniformly random sampling, or arbitrarily random sampling [125, 127, 126]. These strategies do not consider the video content or domain knowledge, leading to low utilization of the limited manual annotation budget. Instead,

in our HVSA framework, we propose Active Sample Selection (ASS) method, which takes both video content and semantic segmentation network into consideration. In detail, we evaluate the prediction uncertainty of the network and try to select samples with least prediction confidence. Also, we generate features of all samples and try to select the most representative samples. In this way, our ASS method not only samples by uncertainty but also by diversity, so it is able to improve the utilization of manual annotation budget and boost the label propagation accuracy.

Curious readers may find that we are doing active *sample* selection, rather than active *frame* selection as in previous work. This is because one of the critical considerations in semantic segmentation is the granularity of the annotation unit. It has been studied in the image semantic segmentation tasks, including frame-based [128, 129, 130], rectangle-based [131, 132, 133], and super pixel-based [134, 135] work. The recent work [135] suggests that super pixel-based annotation is the most efficient for image segmentation tasks. In our ASS method, the sample can be a frame, a rectangle of frame, or even a super pixel. Moreover, to resemble real-world manual annotation process, we first adopt the click-based annotation measurement [131, 133] to simulate annotation cost, then generate "manual annotations" based on clicks and use them in the evaluation. Our experiment results show that optimal granularity in video annotation task is not determined but depends on desired level of annotation quality.

Traditional LP methods [124, 125, 126] propagate manual annotations of selected frames to the entire video only using spatial-temporal information. Therefore, they do not take advantage of semantic information captured in existing semantic segmentation models or manual annotations, leading more manual annotations to fill where the spatial-temporal constraints do not cover. In Test-time Fine-Tuning (TFT) method of our HVSA framework, we design a new loss function considering both spatial-temporal consistency and semantic information in model fine-tuning. It further improves label propagation

quality and saves annotation cost.

In summary, our contributions include:

1. A novel human-in-the-loop framework HVSA, alternating between active sample selection and test-time fine-tuning methods, is proposed for video semantic segmentation auto-annotation at a low annotation cost.
2. In active sample selection, the sample can be a frame, a rectangle of frame, or even a super-pixel. And samples are selected by both uncertainty of the network and the diversity among samples, taking advantage of information from both network and video.
3. In test-time fine-tuning, we propose a new loss function combining both the semantic knowledge and the spatial-temporal information.
4. We study the desired granularity for the video semantic segmentation auto-annotation problem. Our results give insights to the future work along the line in terms of selecting the annotation unit.
5. Real-world experiments, e.g., Figure 5.1, demonstrate that our method generates highly accurate and consistent semantic segmentation annotations of the whole video at a low annotation cost.

5.2 Related Work

In this section, we briefly summarize related work.

Video semantic auto-annotation. Pseudo-labeling and semi-supervised learning are the two popular types of methods for automating video semantic segmentation annotations. The pseudo-labeling approaches [136] use a pre-trained teacher model to generate labels for the test video sequences. However, these approaches are typically frame-based and do not consider the rich temporal constraints in the videos. Therefore, the pseudo-

labels are inevitably noisy, especially when the pre-trained model is trained with the data from a different domain from the input video. This work proposes a novel test-time fine-tuning method to adapt the pre-trained model to the specific video to generate pseudo labels more accurately.

Among the semi-supervised learning approaches, Label Propagation (LP) is widely adapted [125, 127, 137, 138, 139]. Most work uses optical flow to guide the LP process. These methods rely on accurate optical flow estimation, which is difficult to obtain. Otherwise, the erroneous flow estimation can result in propagated labels that are misaligned with their corresponding frames. Rather than conducting direct LP, our method uses the optical flow to generate consistency constraints as a loss to fine-tune the segmentation model, which makes it more robust to flow noise. Moreover, the proposed fine-tuning considers both semantic and temporal information to predict temporally consistent semantic annotations across the full video without the limitations of traditional LP methods. Instead, our test-time fine-tuning is optimizing a new loss that takes both semantic and temporal information into consideration and predicts temporally consistent semantic annotations across the full video without the limitations of traditional LP methods.

Active learning. Rooted in traditional machine learning, active learning [140] allows learners to *actively* query the specific labels they want to obtain, saving labeling costs dramatically. Inspired by the success of active learning, previous methods [141, 134] studied how to select instances to refine a network for segmentation tasks. Our framework's objective is different from them, as we are querying samples from a video such that their annotations could boost the label propagation accuracy on the input video. There is one work [142] studies the active frame selection problem for label propagation. Our work is different in two ways: First, the method in [142] selects frames for just once, while our method could select video frames, rectangles of frames, or even super-pixels in

a human-in-the-loop manner for multiple iterations. Second, the existing method closely ties with a particular LP technique and does not comply with modern deep networks. Our method is generic and can work with different segmentation networks.

Human-in-the-loop for visual annotations. There exists some work [143, 144] trying to reduce the annotation cost in human-in-the-loop model learning. And [145, 146] studied the interactive video object segmentation frameworks. However, solving video semantic segmentation problem in the human-in-the-loop framework has never been studied.

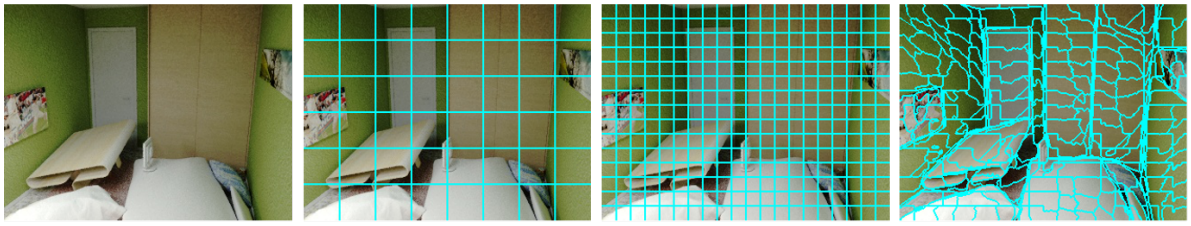
5.3 Methods

In this section, we describe our HVSA framework (Figure 5.2) in detail, including pre-processing, active sample selection, test-time fine-tuning, and cost calculation.

5.3.1 Pre-Processing

Granularity of samples. A suitable sample granularity needs to be carefully chosen to minimize the human annotation effort. We investigate three types of annotation unit: *frame*, *rectangle*, and *super-pixel*, which are typically used in image semantic segmentation tasks. Figure 5.3 shows example of three units. To get rectangle units, we uniformly crop each frame to non-overlapping rectangles. And we use DMMSS-FCN [147] to generate super-pixel units. The n-th sample from the t-th frame is denoted as s_t^n. For frame samples, n is always 0. All samples are prepared in the unlabeled sample pool at the beginning of our framework.

Build temporal correspondence. We rely on correspondences between frames to leverage video temporal information. Here we extract the dense correspondences by estimating optical flow [148], $O_{t\to t'}$, of a frame pair from frame t to t'. Computing



(a) Frame (b) Rec100 (c) Rec16 (d) SP

Figure 5.3: When annotating a video, user could annotate samples of frame (a), rectangle (b) (c), or even super-pixel [147] (d). Segments in (b) and (c) are of size 100 and 16 respectively. There are similar number of segments in (c) and (d).

optical flow for all frame pairs is expensive, thus we limit the distance between frames to be smaller than 3. We further apply a forward-backward consistency check to cope with occlusion/dis-occlusions to extract only reliable correspondences. As a result, each optical flow $O_{t\to t'}$ will have a binary mask $M_{t\to t'}$, where pixels with forward-backward flow difference larger than 1 pixel are set to 0.

5.3.2 Active Sample Selection

To reduce annotation cost, we propose the active sample selection (ASS) to *actively* select the most *important* samples for manual annotations in each iteration. The ASS method takes both the network and the video content into consideration, which involves uncertainty sampling and diversity sampling and their combination.

Margin of confidence and uncertainty sampling. The motivation behind uncertainty sampling is that if a network predicts on a sample with little confidence, this sample needs to be selected for manual annotation. To capture confidence, we use the margin of confidence [8]. For each pixel, its margin of confidence is defined as the difference between the prediction scores of top-1 and top-2 label predictions from the model trained in each iteration. Intuitively, large margin means large prediction confidence. After being subtracted from 1, the pixel margin of confidence is converted to the pixel

uncertainty. The uncertainty of sample s_t^n is then defined as the summation of pixel uncertainties within the sample region:

$$u(s_t^n) = \sum_{x \in s_t^n} P_{\theta_{k-1}}(y_{1,x}^* | I_t) - P_{\theta_{k-1}}(y_{2,x}^* | I_t), \tag{5.1}$$

where I_t is the input frame, y^* is the prediction from $softmax$, x is a pixel position within s_t^n, and θ_{k-1} is the previous model. By applying uncertainty sampling, the ASS method knows what are the samples that the current network is unsure about its prediction and then theses samples will be selected accordingly.

However, uncertainty sampling has a shortcoming in isolation. It might focus on one part of the decision boundary and select similar samples, causing a waste of human effort. To make the selection strategy comprehensive, we further require the method to samples that are different from each other, which refers to the *diversity sampling*.

Deep feature and diversity sampling. Clustering-based sampling naturally targets a diverse selection of samples. We first conduct clustering on unlabeled samples then select centroid samples for annotation. We re-use the downstream segmentation model as a feature extractor. Specifically, we transform each frame I_t to a feature map F_t using the previous model backbone network without segmentation head. Then the sample feature $\mathbf{f}_t^n$ is defined as the average along the spatial dimensions of F_t within s_t^n region:

$$\begin{aligned} F_t &= \psi_{\theta_{k-1}}(I_t), \\ \mathbf{f}_t^n &= \text{MeanPool}_{x \in s_t^n}(F_{t,x}), \end{aligned} \tag{5.2}$$

where ψ denotes the segmentation network backbone. We employ the k-Means algorithm with Euclidean distance on $\mathbf{f}$ for clustering.

Combining uncertainty and diversity sampling. In first iteration of our framework, as the network hasn't been fine-tuned, we only apply diversity sampling. Later

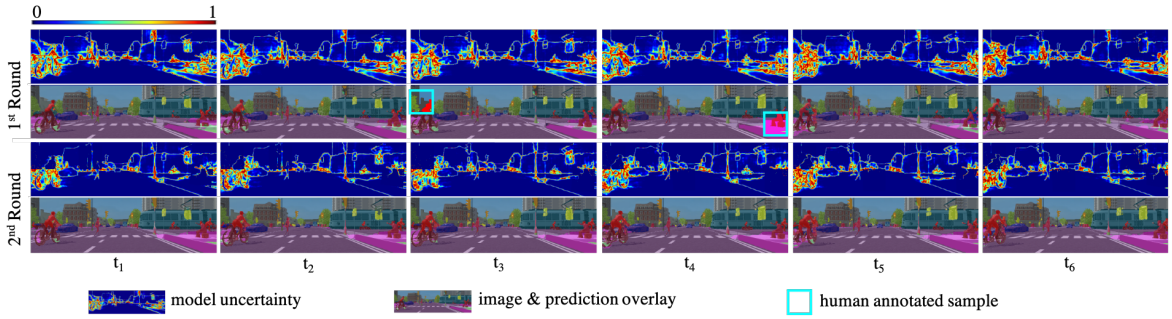


Figure 5.4: Visualization of model uncertainty and annotation selections on VEIS. After fine-tuning on the high uncertainty sample, the generated annotation in the second iteration improves significantly within sample regions across all neighbor frames.

in iterations, we first selects half of the most uncertain samples and cluster them into b clusters, where b is the annotation budget in one iteration. Then, b cluster centroids are selected and sent to human annotators. In this way, selected samples are of high uncertainty and are relatively different from each other. See Figure 5.4 for an example.

5.3.3 Test-time Fine-tuning on Input Video

While a network may be pre-trained on relevant datasets, directly applying it to an arbitrary video would lead to inferior results, e.g., Figure 5.5. To progressively adapt it to a video, in each iteration, we fine-tune the model leveraging two different information sources, inspired by how human annotators handle the video annotation tasks. Given a target frame and the video, an annotator will naturally analyze its neighbor frames to decide the correct categories of the objects in the scene; The annotator will also refer to the existing annotations within the same video. Moreover, we propose a new loss designed from the two information sources, and show how we optimize it.

Temporal consistency loss. Our *temporal consistency loss*, $\mathcal{L}_{tc}$, encourages consistent predictions across corresponding pixels on different frames. Unlike other methods [149, 126] which directly propagate labels between frames, we propagate predicted

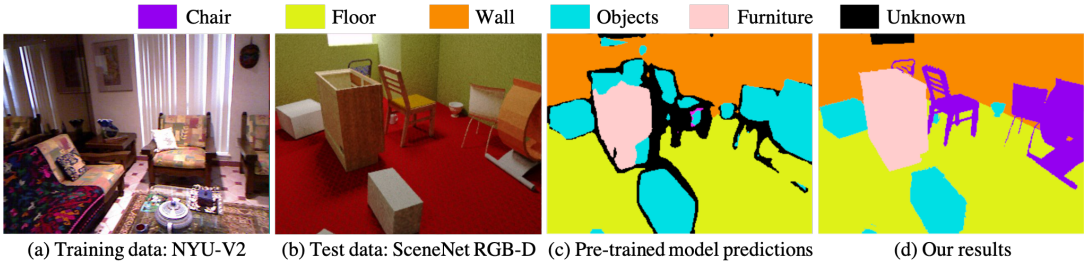


Figure 5.5: The model pre-trained on NYU-V2 performs poorly on new out-of-domain input video as in (c). (d) Our framework adapts the model to the input video and produces better results.

class probabilities from the model. More specifically, we penalize the difference between predicted class probabilities $\mathbf{q}_t$ and $\mathbf{q}'_t$ of frame t and t' at pixel position x as:

$$\mathcal{L}_{tc,t\to t'}(x) = M_{t\to t'}(x)\,\|\mathbf{q}_t(x) - \hat{\mathbf{q}}_{t'\to t}(x)\|_2^2\,, \tag{5.3}$$

where $\hat{\mathbf{q}}_{t'\to t}(x)$ is the warped prediction score from frame t' to t using the pre-computed flow $F_{t\to t'}$ and $M_{t\to t'}$ is the mask associated with $F_{t\to t'}$.

Here we illustrate why and how we have the mask $M_{t\to t'}$. We apply a forward-backward consistency check to cope with occlusion/dis-occlusions to extract only reliable correspondences. As a result, each optical flow $O_{t\to t'}$ will have a binary mask $M_{t\to t'}$, where pixels with forward-backward flow difference larger than 1 pixel are marked as 0. $M_{t\to t'}$ at position x can be formulated as

$$M^{(x)}_{t\to t'} = \mathbf{1}\left[\left\|O^{(x)}_{t\to t'} - \hat{O}^{(x)}_{t'\to t}\right\|_2^2 < 1\right], \tag{5.4}$$

where $\hat{O}_{t'\to t}$ is the warped version of $O_{t'\to t}$ using flow $O_{t\to t'}$. So that the position of $O_{t\to t'}$ and $\hat{O}_{t'\to t}$ is aligned and they can be compared directly.

Unlike the existing approaches [149] which only consider the temporal relation between annotated frames and their neighbors, we apply the temporal consistency loss to

even unlabeled image pairs. As a result, labeled image information transforms to more than 3 frames away, which is the distance limitation of optical flow.

Semantic loss. Temporal constraints tell the model which pixel to share labels with but not where to hold. This semantic information will have to come from the annotated samples on the input video. We compute the regular cross-entropy loss, $\mathcal{L}_{ce}$, for any frame or frame region with manual annotations:

$$\mathcal{L}_{ce,t} = \mathcal{L}_{CE}(\mathbf{q}_t, L_t), \tag{5.5}$$

L_t denotes the semantic label at frame t, where unlabeled region is set to a special "ignored index".

Optimization. In the test-time fine-tuning, each training sample consists of two frames that pass through the single-frame model in parallel, giving two sets of class probability predictions. The two predictions are then used to compute the temporal loss $\mathcal{L}_{tc}$. If any frame region of the pair has manual annotations, the cross-entropy loss $\mathcal{L}_{ce}$ will be calculated as well. In summary, we fine-tune the single-frame segmentation network weights using standard backpropagation during test-time fine-tuning by minimizing:

$$\mathcal{L} = \lambda \mathcal{L}_{tc} + \mathcal{L}_{ce}. \tag{5.6}$$

We initialize the network weights using the pre-trained model in the first selection iteration. In later iterations, the network fine-tunes from the previous checkpoint, and then predicts segmentation labels on all the frames.

5.3.4 Annotation Cost Calculation

In practice, the annotation cost is measured by expense or human labeling time. Some conventional semantic segmentation AL work [134] uses percentage of labeled pixels to represent manual effort. We follow some recent work [131, 133, 135] to measure cost by annotation clicks, which is more realistic. Semantic segmentation label mask is pixel-level, while in actual labeling tasks, human annotators usually use a polygon-based tool [133]. Annotators first click on several vertices on the boundary of the one object to form a closed polygon ("Boundary click"), then select the object type by clicking once ("Class click"). In this way, all pixels within this polygon get the label of this class.

Here we introduce how we use algorithm to mimic human annotator to locate the "Boundary click" positions from the existing segmentation labels, and calculate the total clicks as the annotation cost. For each connect component of a single class object, we find its contour pixels, and simplify the contour pixels into some polygon vertices using Ramer–Douglas–Peucker (RDP) algorithm. Each polygon vertex costs one "Boundary click". In addition, each polygon costs one "Class click" to specify its class label. Figure 5.6 shows an example.

For rectangle-based and super-pixel-based annotations, there are no clicks required on the sample boundary. If a sample only consists of a single class object, the required number of clicks is one "Class click". For super-pixel, unlike [135], we do not assign the dominant label to the entire super-pixel since the error label will be propagated to neighbor frames, hurting the final annotation quality.

Mimic "manual annotation". [135] uses a similar method to estimate annotation clicks, while using the GT labels provided by the dataset as training labels. However, this is not appropriate, as the RDP algorithm simplifies the object polygon boundaries, which leads to a rougher annotation of GT. In their case, the click-based cost is underestimated

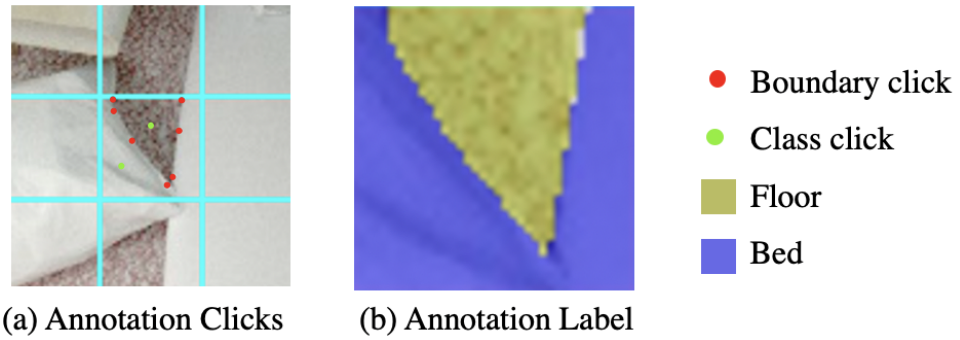


Figure 5.6: Example of annotating the center rectangle sample in (a). The red "Boundary click" are generated by the RDP algorithm from the original object contour. No clicks are needed in the boundaries of the sample to enclose the polygon. The green "Class click" specifies Bed and Floor class in this example. (b) is the segmentation label annotated by the shown 9 clicks.

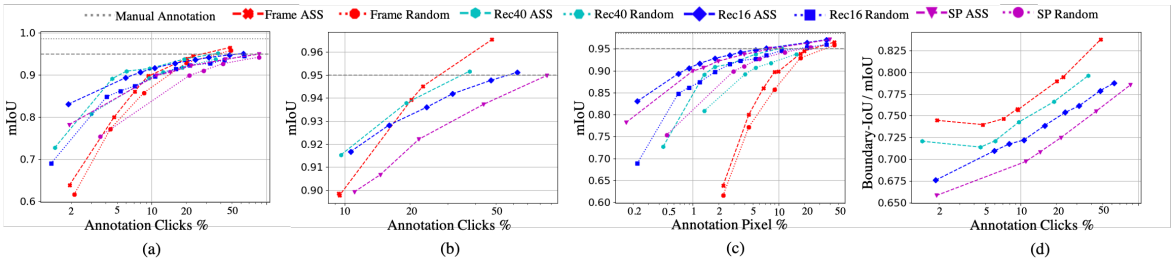


Figure 5.7: Active sample selection results on SceneNet RGB-D: (a),(b),(d) show the generated annotation mIoU and the normalized boundary-IoU in annotation clicks %, and (b) is a zoom-in version of (a). (c) shows the generated annotation mIoU in annotated pixel %.

compared to the training label quality. On the contrary, we mimic manual annotation (MA) by converting the simplified polygons back to label masks. We use MA rather than GT in fine-tuning models, which better fits the video segmentation annotation tasks in practice.

5.4 Experiments

In this section, we conduct experiments on two datasets with dense segmentation GT on every frame to support the evaluation of the framework. We first compare the proposed

ASS method with different frame selection baselines using various sample granularities. Then, we study the effectiveness of the proposed test-time fine-tuning by comparing it with other label propagation methods. Finally, we envision the generated annotations to show more details of the outputs from the proposed framework.

5.4.1 Experimental Settings

Training settings

We perform three iterations of ASS for each testing sequence. The annotation budget for each iteration is divided equally from the total budget.

We use the HRNet-W48 [150] as the backbone network (other networks can be easily incorporated). We set the consistency loss weight $\lambda = 1$. The initial learning rate in each iteration is 0.004. In each iteration, we fine-tune the network for 15 epochs with a learning rate of 0.004 and SGD optimizer [151] with momentum 0.9. We follow the "poly" learning rate policy to reduce the learning rate gradually. The batch size is 14 for SceneNet RGB-D [152] dataset, and 2 for VEIS [153] dataset.

In the ASS diversity sampling, we transform each frame to feature space using the previous iteration backbone network without segmentation head:

$$F_t = \psi_{\theta_{k-1}}(I_t), \tag{5.7}$$

where ψ denotes the segmentation network backbone. More concretely, the feature map F_t is the concatenation of four feature maps after the fourth stage of HRNet-W48. The number of channels of F_t is 720.

We use the RAFT-things checkpoint [148] to generate the flow correspondence and use SpixelFCN-bsd checkpoint [147] to generate the super-pixels. All the settings in

RAFT and SpixelFCN are as default. We use openCV [154] connectedComponents, findContours, and approxPolyDP functions to estimate the "Boundary Click" positions. The parameter ϵ in approxPolyDP controlling the fineness of simplified polygons is set to 1.

For HRNet backbone, RAFT and SpixelFCN, we use the code from their official implementation in Pytorch [155]. All the experiments run on machines with 4×Nvidia 1080s.

Evaluation and metrics

We use four metrics to evaluate our method thoroughly, which are pixel accuracy, mean Intersection over Union (mIoU), boundary Intersection over Union (Boundary-IoU), and temporal consistency. The first two are commonly used in segmentation tasks to measure the accuracy of predictions. Here we only illustrate the last two metrics in detail.

Boundary-IoU. The boundary-IoU is proposed in [156]. The boundary-IoU between our prediction $\mathbf{Q}$ and ground-truth $\mathbf{G}$ is calculated as:

$$\text{boundary-IoU}\left(\mathbf{G}, \mathbf{Q}\right) = \frac{|(\mathbf{G}_d \cap \mathbf{G}) \cap (\mathbf{Q}_d \cap \mathbf{Q})|}{|(\mathbf{G}_d \cap \mathbf{G}) \cup (\mathbf{Q}_d \cap \mathbf{Q})|}, \tag{5.8}$$

where $\mathbf{G}_d$ and $\mathbf{Q}_d$ are the sets of pixels in the boundary region of the ground-truth mask and the prediction mask respectively. d is the pixel width of the boundary region. We set d to 2 in all experiments.

Temporal consistency (TC). We also measure the temporal consistency (TC) of the generated annotations by measuring the mIoU between two consecutive predictions similar to [157].

We measure the temporal consistency (TC) of the generated annotations by measuring

the mIoU between two consecutive predictions similar to [157]. The TC between frame t and frame $t-1$:

$$TC\left(\mathbf{Q}_{t-1}, \mathbf{Q}_t\right) = \frac{|\mathbf{Q}_t \cap \hat{\mathbf{Q}}_{t-1}^{(M)}|}{|\mathbf{Q}_t \cup \hat{\mathbf{Q}}_{t-1}^{(M)}|}, \tag{5.9}$$

where $\mathbf{Q}_t$ is the prediction of frame t, $\mathbf{Q}_{t-1}$ is the prediction of frame $t-1$. $\hat{\mathbf{Q}}_{t-1}^{(M)}$ is the warped prediction from frame $t-1$ to frame t, and pixels where does not pass the forward-backward check $M_{t-1\to t}$ will be marked as ignored label in $\hat{\mathbf{Q}}_{t-1}^{(M)}$. The calculation of $TC\left(\mathbf{Q}_{t-1}, \mathbf{Q}_t\right)$ is very similar with calculating the standard IoU in the segmentation task, where we treat prediction as $\mathbf{Q}_t$, and ground truth as $\hat{\mathbf{Q}}_{t-1}^{(M)}$. So the TC on all the test sequences can be calculated similar to IoU and mIoU on the whole test set in a segmentation task.

5.4.2 Comparative Assessment

SceneNet RGB-D

We use the SceneNet RGB-D [152], which is a photorealistic indoor trajectory dataset with semantic segmentation annotations for every video frame to evaluate the overall system performance. Unlike regular indoor scene datasets [158, 159], the room layouts/object placements of the ScenNet RGB-D dataset are generated randomly. We train a 14-class HRNet-W48 model using the NYU-V2 [159] training set as the pre-trained model, which has only 15.04% mean-Intersection-over-Union (mIoU) on the SceneNet testing videos. We will demonstrate that our test-time fine-tuning method adapts the segmentation model to randomly generated scenes and achieves more satisfying results (examples in Figure 5.5). We randomly picked five sequences from the SceneNet test set in our experiments, each containing 300 frames. We test four granularity settings: frame, 40×40-pixel rectangle, 16×16-pixel rectangle and super-pixel, denoted as Frame, Rec40,

Rec16, and SP respectively. Given the SceneNet frame resolution of 240×320, Rec40 and Rec16 split a frame into 56 and 300 segments respectively. We let SP split a frame into about 300 segments.

We evaluate the generated annotations by measuring their mIoU with the GT. Figure 5.7 compares the generated label quality from different selection methods and annotation sample granularities. The "Annotation Clicks %" (shown in log scale) is the number of annotation clicks normalized by the number of clicks to annotate the whole video. We can see from (a) that the proposed ASS method outperforms random selection baselines in all sample granularity. Rec16 gives the best annotation mIoU with fewer clicks among all the granularities because it provides better sample diversity than larger samples. This diversity favors model fine-tuning when annotations are limited. As annotation clicks increase, the gap between all the settings becomes smaller, so we zoom in on the curves in this part in (b).

Annotating Frame surpasses others when the percentage of clicks is over 20%. The reason is that the sample diversity saturates with more manually annotated samples. In this stage, annotating more pixels keep improving final outputs quality by label propagation. Annotating frames obtains the most labeled pixels per click compared to smaller-sized samples, due to the effort to handle truncated object contours or the dividing objects merged by imperfect super-pixels. As a result, a larger granularity annotation sample achieves higher label quality faster. To this end, we suggest users choose a proper sample granularity to annotate depending on their desired label quality.

Click cost for label quality benchmarks. In Table 5.1 we list the least annotation clicks required to generate 80%, 85%, 90%, and 95% mIoU labels, and the corresponding sample granularity. The last row represents the manual labeling of the full video. Annotating Rec16 samples first achieves 80% and 85% mIoU, the annotation click cost is 1.5% and 2.5%. Rec40 first achieves 90% mIoU with 5% of annotation clicks. Annotating

Table 5.1: This table shows the most efficient sample granularity for different mIoU benchmarks in SceneNet. The last row represents manually annotating all the frames.

Annotation mIoU	Granularity	Anno. Clicks	Anno. Pixel
80%	Rec16	1.5%	0.2%
85%	Rec16	2.5%	0.3%
90%	Rec40	5.0%	1.4%
95%	Frame	27%	23%
99%	Frame	100%	100%

Frame first achieves 95% mIoU with 27% annotation clicks, which is over five times the clicks to achieve 90% mIoU. This observation shows the mIoU gain is sub-linear to the annotation clicks. However, it still saves 73% annotation effort compared to annotating the full video, demonstrating the proposed method generates very high-quality annotations while saving human effort significantly. It is worth mentioning that the pre-trained model performs poorly on testing sequences (Figure 5.5), which shows the proposed framework can adapt to the target sequence by learning from selected samples and leveraging the temporal information.

Figure 5.7 (c) shows the comparison under the traditional pixel-based annotation cost measurement. The observation is very different from (a), as annotating Frame is always the worst. We believe that the traditional pixel-based cost measurement could be misleading in Segmentation AL tasks.

Comparison of boundary-IoU. Object boundary quality is crucial in segmentation annotations. In Figure 5.7 (d), we show the boundary-Intersection-over-Union (boundary-IoU) [156] normalized by mIoU, which reflects the boundary annotation accuracy. Models trained on frame samples outperform the others with no exceptions. The reason is frame level annotation provides the richest semantic/boundary information. On the contrary, the super-pixel-based selection is usually composed of pixels of the same object, which

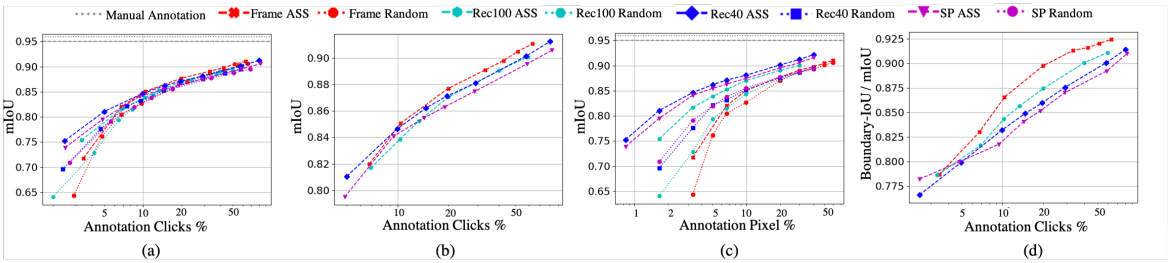


Figure 5.8: Active sample selection results on VEIS: (a),(b),(d) show the generated annotation mIoU and the normalized boundary-IoU in annotation clicks %, and (b) is a zoom-in version of (a). (c) shows the generated annotation mIoU in annotated pixel %.

Table 5.2: Comparison of the overall performance on SceneNet [152] with manual annotations selected by ASS method. Given the same information from annotated frames, our method outperforms the other two and shows advantages at lower annotation cost.

	2% clicks			4.6% clicks			7.1% clicks			9.3% clicks		
	mIoU	P-Acc.	TC	mIoU	P-Acc.	TC	mIoU	P-Acc.	TC	mIoU	P-Acc.	TC
Fine-tune only	45.72	76.57	61.98	64.73	88.90	76.86	70.76	91.76	81.20	81.31	94.72	87.09
LP [126]	48.57	76.46	66.09	59.5	85.06	84.68	68.2	87.66	86.97	76.34	91.41	86.66
Ours	**63.67**	**88.71**	**84.43**	**79.54**	**95.15**	**89.33**	**86.07**	**96.90**	**93.45**	**89.96**	**97.28**	**94.70**

lacks the information of the object boundaries. So its boundary prediction accuracy is the worst. For rectangle samples, larger granularity samples give better predictions on the boundary. If the user has high requirements on the label boundary quality, annotating whole frames is the best choice.

VEIS

For more extensive experiments, we conducted auto-annotation experiments on an outdoor-scene synthetic dataset VEIS [153]. It includes semantic segmentation ground-truth for every video frame with the object classes of standard real urban scene datasets, such as CamVid [123] and Cityscapes [122]. We randomly pick six video clips from the full VEIS sequence, each of which contains 200 frames. The pre-trained model is trained with Cityscapes training set from an ImageNet pre-trained checkpoint with mIoU of

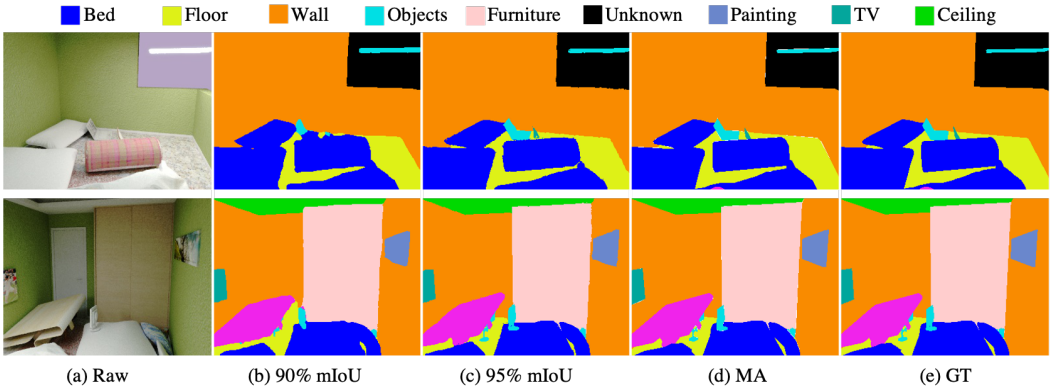


Figure 5.9: Visualization of our generated annotations in SceneNet RGB-D. (a) is the video frame, (b) costs about 5.0% clicks with annotating Rec40, and (c) costs about 27% clicks with annotating Frame. (d) is the mimic manual annotation, and (e) is the ground-truth.

32.56% on all the testing videos. We tested four granularity settings: Frame, Rec100, Rec40, and SP. As the resolution of VEIS frames is 600×800, Rec100 and Rec40 split a frame into 48 and 200 segments, respectively. We let SP split a frame into about 200 segments.

Figure 5.8 (a) compares the generated label mIoU given annotation clicks, and (b) zooms in the high mIoU plots. The observations are very similar to the SceneNet results. First, the ASS method always outperforms random selection baselines. Second, larger granularity annotation samples achieve higher label quality faster. When annotation clicks are small, annotating Rec40 samples leads to the best-generated annotations. After the annotation cost in clicks is greater than 10%, annotating Frame outperforms all others.

5.4.3 Analysis

Model uncertainty and selected samples. Figure 5.4 illustrates how the proposed framework selects sample and learns from it. This VEIS example is of annotating Rec100 with about 3.3% annotation clicks. The first two rows are the model uncertainty and

generated annotation after the first iteration. The ASS method selects a sample that are of high uncertainty and inferior prediction. The similar regions in other frames are not selected, as the proposed ASS considers both sample uncertainty and diversity. The last two rows show the model results after fine-tuning with the selected sample. The regions' annotation quality in all the neighbor frames is improved significantly, demonstrating the effectiveness of the test-time fine-tuning component.

Effectiveness of label propagation module. We compare the proposed test-time fine-tuning method with it's ablated version by removing temporal consistency loss (Fine-tune only) and LP [126]. LP is a well known label propagation algorithm, which can be directly applied to new target domain videos to propagate sparse annotations. Here we use our ASS method to select manual annotated samples. Table 5.2 shows generated label's mIoU, pixel accuracy, and Temporal consistency (TC). TC measures the mIoU between two consecutive predictions similar to [157]. Given the same selected samples, our method outperforms the Fine-tune only and LP methods by a large margin in mIoU and TC at various annotation clicks percentages. The results prove the effectiveness of consistency loss and test-time fine-tuning method. The benefit is even more significant when the sample rates are lower, as our method incorporates both motion and semantic cues to the test sequences.

Impact of number of ASS iteration. We conduct experiments to understand the impact of the number of iterations to the segmentation quality on SceneNet RGB-D. We feed 0.3% clicks of annotations per iteration, and fine-tune the model up to nine iterations. The mIoU gains per iteration are 6.91%, 1.32%, 0.89%, 0.35%, 0.16%, 0.43%, 0.07%, 0.31%, and 0.03%. Starting from the fourth iteration, the mIoU gain becomes negligible. As a result, we use three iterations for ASS.

Error pattern in high quality generated annotations. We conduct experiments to investigate where the remaining errors are when the generated annotation is already of

high quality. On SceneNet RGB-D, the 100% manual annotation mIoU is 98.56%. When our method achieves 97.46% mIoU, the boundary IoU is only 83.44%, indicating errors appear in the object boundaries. Categories with high boundary-to-area ratio have the largest impact from the imperfect boundary predictions. This can be reflected from their below-average per-class IoU. In SceneNet, they are "Object", "Chair", and "Table". In VEIS, they are "Pole", "Traffic Light", and "Rider". With the error pattern in mind, users could use the generated annotations more confidently.

Generated annotation visualization. In Figure 5.9 we show our generated annotations in SceneNet. The 90% mIoU annotations in (b) only cost about 5.0% clicks; The 95% mIoU annotations in (c) cost about 27% clicks.

Model computation time. The model computation time for one ASS iteration is mainly from sample selection and test-time training steps. The test time fine-tuning computation time depends on image resolution and video sequence length. For SceneNet RGB-D, a sequence of 300 frames with resolution 320×240 takes 20 minutes for one iteration on average. For VEIS, a sequence of 200 frames with resolution 800×600 takes 33.3 minutes for one iteration on average. Our experiments runs on 4×Nvidia 1080s. The 9 seconds sample selection CPU runtime can be neglected. The dozens of minutes computation time prevents the annotators from labeling the next batch of samples immediately. However, this can be easily mitigated by multitasking arrangements in practice.

Table 5.3: This table shows the most efficient sample granularity for different mIoU benchmarks in VEIS. The last row represents manually annotating all the frames.

Annotation mIoU	Granularity	Anno. Clicks	Anno. Pixel
80%	Rec40	4.3%	1.3%
85%	Frame	10.4%	10%
90%	Frame	44%	51%
96%	Frame	100%	100%

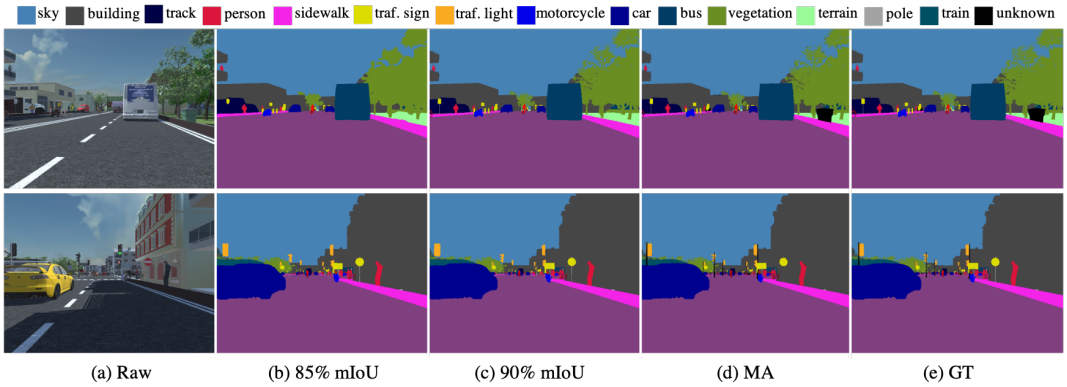


Figure 5.10: Visualization of our generated annotations in VEIS. (a) is the video frame, (b) costs about 10.4% clicks with annotating Frame, and (c) costs about 44% clicks with annotating Frame. (d) is the mimic manual annotation, and (e) is the ground-truth.

Click Cost for Label Quality Benchmarks. Table 5.3 lists the least annotation clicks required to generate 80%, 85%, and 90% mIoU labels, and the corresponding sample granularity. Notice that in the VEIS dataset, the mIoU for fully manual annotation is 96%, which is lower than 99% of SceneNet. This difference is mainly because the RDP algorithm simplifies the over-detailed "bicycle" ground-truth in VEIS. As a result, we test up to 90% mIoU annotation quality. Annotating Rec40 makes the model generate 80% mIoU annotations with the least clicks, which is only 4.3%. Annotating frames outperforms other granularity settings when targeting 85% and 90% mIoU annotations. The percentages of clicks are 10.4% and 44% respectively.

We noticed that the percentage of clicks needed in VEIS is much larger than in SceneNet. This is mainly because VEIS images are of higher resolution, and objects include more details. This leads to more clicks in segmentation annotation.

Comparison of Boundary-IoU. In VEIS, the observation is similar to it in SceneNet: Larger granularity samples give better predictions on the boundary, and training on SP gives the worst boundary quality.

Generated annotation visualization. Figure 5.10 shows the visualizations in VEIS.

The 85% mIoU annotations in (b) cost 10.4% clicks, and only details of "tree" are imperfect. The 90% mIoU annotations in (c) cost 44% clicks and are without a significant visual difference to MA.

5.5 Conclusions

We propose a human-in-the-loop framework HVSA to generate video semantic segmentation annotations. It actively selects annotation samples at each iteration that bring the most information for annotating. After selected samples get manual annotations, our method leverages both semantic knowledge and temporal constraints to fine-tune a video-specific semantic segmentation model. Finally, the model is used to generate annotations for the entire video. We conducte experiments on two datasets to show HVSA can generate close-to-perfect annotations at a low cost, even without good pre-trained networks. Each iteration of HVSA takes dozens of minutes, which can be further optimized using multi-task parallelization.

Chapter 6

Conclusions

In this thesis, I presented my research on global black-box optimization, misspecified linear bandits, active learning for privacy protection, and active sample selection for video semantic segmentation, all under the adaptive sequential decision making framework. In the future, there are still many interesting research directions that readers can pursue.

Global black-box function optimization with structured bandit feedback. In material science, sometimes absolute material performance cannot be quantitatively measured, but relative performance can be easily obtained. For example, if mineral A scratches mineral B and mineral B cannot scratch mineral A, then mineral A is harder than B. It motivates the challenging global optimization with comparison feedback problem. Although there is a line of existing work called preferential Bayesian optimization [160] studying this problem, however, its modeling capacity is significantly limited by preferential function being assumed to be logistic function only. Solving a generic global optimization with comparison feedback problem requires substantial novelty and has strong real-world impacts.

Also, modern cutting edge material design usually requires expensive experiments which cannot be conducted without team work. However, if multiple scientists are in-

volved in the same task at the same time, communication between them raises a new challenge. It motivates the distributed global optimization problem where at each round multiple actions are taken and multiple observations are received. The key problem is how to manage observations and assign actions for the next round. Systematically studying this problem will not only guide real-world material design experiments but also provides a foundational understanding for team work in more applications.

Adaptive sequential decision making with domain knowledge. All my research presented in this thesis is purely driven by machine learning and ignores physical and chemical properties if they are available. When the material performance is not a pure black-box function, domain knowledge plays an important role in material design. Therefore, incorporating domain knowledge into the adaptive sequential decision making framework is an important and practical direction in the future.

Adaptive sequential decision making for more applications. Adaptive sequential decision making, as one of the key parts of machine learning, should never be restricted to core machine learning research or limited real-world applications. Future directions include research on more topics such as medicine, health science, computational social science, financial service analytics, and so on.

Appendix A

Auxiliary Technical Lemmas

In this chapter, I list auxiliary lemmas that are used in proofs.

Lemma A.0.1 (Adapted from eq. (5) (6) of [15]) *Given a dataset* $\{x_i, y_i\}_{j=1}^n$ *where* y_j *is generated from eq. (2.1) and* f_0 *is the underlying true function. Let* $\hat{f}$ *be an ERM estimator taking values in* $\mathcal{F}$ *where* $\mathcal{F}$ *is a finite set and* $\mathcal{F} \subset \{f : [0,1]^d \to [-F, F]\}$ *for some* $F \geq 1$*. Then with probability* $> 1 - \delta$*,* $\hat{f}$ *satisfies that*

$$\mathbb{E}[(\hat{f} - f_0)^2] \leq \left(\frac{1+\alpha}{1-\alpha}\right)\left(\inf_{f \in \mathcal{F}} \mathbb{E}[(f - f_0)^2] + \frac{F^2 \log(|\mathcal{F}|)\log(2)}{n\alpha}\right) + \frac{2\log(2/\delta)}{n\alpha},$$

for all $\alpha \in (0, 1]$*.*

Lemma A.0.2 (Self-normalized bound for vector-valued martingales [16, 18]) *Let* $\{\eta_i\}_{i=1}^\infty$ *be a real-valued stochastic process with corresponding filtration* $\{\mathcal{F}_i\}_{i=1}^\infty$ *such that* η_i *is* $\mathcal{F}_i$ *measurable,* $\mathbb{E}[\eta_i | \mathcal{F}_{i-1}] = 0$*, and* η_i *is conditionally* σ*-sub-Gaussian with* $\sigma \in \mathbb{R}^+$*. Let* $\{X_i\}_{i=1}^\infty$ *be a stochastic process with* $X_i \in \mathcal{H}$ *(some Hilbert space) and* X_i *being* F_t *measurable. Assume that a linear operator* $\Sigma : \mathcal{H} \to \mathcal{H}$ *is positive definite, i.e.,* $x^\top \Sigma x > 0$ *for any* $x \in \mathcal{H}$*. For any* t*, define the linear operator* $\Sigma_t = \Sigma_0 + \sum_{i=1}^t X_i X_i^\top$

(here $xx^\top$ denotes outer-product in $\mathcal{H}$). With probability at least $1-\delta$, we have for all $t \geq 1$:

$$\left\| \sum_{i=1}^{t} X_i \eta_i \right\|^2_{\Sigma_t^{-1}} \leq \sigma^2 \log\left(\frac{\det(\Sigma_t)\det(\Sigma_0)^{-1}}{\delta^2} \right).$$

Lemma A.0.3 (Sherman-Morrison lemma [161]) *Let A denote a matrix and b, c denote two vectors. Then*

$$(A + bc^\top)^{-1} = A^{-1} - \frac{A^{-1}bc^\top A^{-1}}{1 + c^\top A^{-1} b}.$$

Lemma A.0.4 (Lemma 6.10 of [18]) *Define $u_t = \sqrt{x_t^\top \Sigma_t^{-1} x_t}$ and we have*

$$\det \Sigma_T = \det \Sigma_0 \prod_{t=0}^{T-1} (1 + u_t^2).$$

Lemma A.0.5 (Potential function bound (Lemma 6.11 of [18])) *For any sequence $x_0, ..., x_{T-1}$ such that for $t < T, \|x_t\|_2 \leq C_b$, we have*

$$\begin{aligned} \log\left(\frac{\det \Sigma_{T-1}}{\det \Sigma_0} \right) &= \log\det\left(I + \frac{1}{\lambda} \sum_{t=0}^{T-1} x_t x_t^\top \right) \\ &\leq d \log\left(1 + \frac{T C_b^2}{d\lambda} \right). \end{aligned}$$

Lemma A.0.6 (Pointwise convergence [107]) *Let (x, z) be drawn from any distribution $\mathcal{D}$ supported on $\mathcal{X} \times \mathcal{Y}$. Let* Dis *and* $\widehat{\text{Dis}}$ *be the expected and empirical disagreement evaluated on n i.i.d. samples from $\mathcal{D}$. For each fixed $h \in \mathcal{H}$, the following generalization error bound holds with probability $1-\gamma$,*

$$\text{Dis}(h, z) \leq \widehat{\text{Dis}}(h, z) + \sqrt{\frac{2\text{Dis}(h, z)\log(1/\gamma)}{n}} + \frac{2\log(1/\gamma)}{3n},$$

where n is the number of data points.

This is a standard application of the Bernstein's inequality.

Lemma A.0.7 (Uniform convergence [107]) *Under the same conditions of Lemma A.0.6, and in addition assume that d is the VC-dimension of $\mathcal{H}$, Then with probability at least $1-\gamma$, $\forall h \in \mathcal{H}$ simultaneously,*

$$\mathrm{Dis}(h,z) - \widehat{\mathrm{Dis}}(h,z) \leq 2\sqrt{\frac{(d+\log(4/\gamma))\widehat{\mathrm{Dis}}(h,z)}{n}} + \frac{4(d+\log(4/\gamma))}{n}.$$

and

$$\mathrm{Dis}(h,z) - \widehat{\mathrm{Dis}}(h,z) \leq 2\sqrt{\frac{(d+\log(4/\gamma))\mathrm{Dis}(h,z)}{n}} + \frac{4(d+\log(4/\gamma))}{n}.$$

The above lemma is simply the uniform Bernstein's inequality over a hypothesis class with VC-dimension d. We will be taking z to be h^* in the cases when we work with noise conditions and $h_\infty^{\mathtt{agg}}(x)$ in the agnostic case.

Lemma A.0.8 (Learning bound [107]) *Let d be the VC-dimension of $\mathcal{H}$, the excess risk is bounded with probability $1-\gamma$,*

$$\mathtt{Err}(\hat{h}) \leq \mathtt{Err}(h^*) + 2\sqrt{\mathtt{Err}(h^*)\frac{d\log(n)+\log(4/\gamma)}{n}} + 4\frac{d\log(n)+\log(4/\gamma)}{n},$$

where n is the number of data points we sample.

Lemma A.0.9 (Passive learning bound under TNC (Lemma 3.4 of [93])) *Let d be the VC-dimension of the class $\mathcal{H}$. Assume Tsybakov noise condition with parameters*

τ, the excess risk is bounded with probability $1-\gamma$,

$$\texttt{Err}(\hat{h}) - \texttt{Err}(h^*) \lesssim \left(\frac{1}{n}\left(d\log\left(\frac{n}{d}\right) + \log\left(\frac{1}{\gamma}\right)\right)\right)^{\frac{1}{2-\tau}},$$

where n is the number of data points.

Lemma A.0.10 (Agnostic active learning bound (Theorem 5.4 of [93])) *Let $\mathcal{H}$ be a class with VC-dimension d. With probability at least $1-\gamma$, there is a universal constant C, such that the agnostic active learning algorithm (see Algorithm 7) outputs a classifier with an access risk of α with*

$$C\theta(\texttt{Err}^* + \alpha)\left(\frac{(\texttt{Err}^*)^2}{\alpha^2} + \log\left(\frac{1}{\alpha}\right)\right)\left(d\log(\theta(\texttt{Err}^* + \alpha)) + \log\left(\frac{\log(1/\alpha)}{\gamma}\right)\right),$$

where $\texttt{Err}^* = \operatorname{argmin}_{h\in\mathcal{H}} \texttt{Err}(h)$.

Appendix B

Additional Information about Differential Privacy

In this chapter, I cite a few results from differential privacy that I use as part of the analysis.

Lemma B.0.1 (Post-processing [75]) *If a randomized algorithm $\mathcal{M} : \mathcal{Z}^* \to \mathcal{R}$ is (ϵ, δ)-DP, then for any function $f : \mathcal{R} \to \mathcal{R}'$, $f \circ \mathcal{M}$ is also (ϵ, δ)-DP.*

Definition B.0.2 (Global sensitivity [104]) *A function $f : \mathcal{Z}^* \to \mathcal{R}$ has global sensitivity ϑ if*

$$\max_{|D-D'|=1} \|f(D) - f(D')\|_1 = \vartheta.$$

Lemma B.0.3 (Laplace mechanism [75]) *If a function $f : \mathcal{Z}^n \to \mathcal{R}^p$ has global sensitivity ϑ, then the randomized algorithm $\mathcal{M}$, which on input D outputs $f(D) + b$, where $b \sim \texttt{Lap}(\vartheta/\epsilon)^p$, satisfies ϵ-DP. The $\texttt{Lap}(\lambda)^p$ denotes a vector of p i.i.d. samples from the Laplace distribution $\texttt{Lap}(\lambda)$.*

Definition B.0.4 (ℓ_2-sensitivity [104]) *A function $f : \mathcal{Z} \to \mathcal{R}$ has ℓ_2 sensitivity ϑ_2 if*

$$\max_{|D-D'|=1} \|f(D) - f(D')\|_2 = \vartheta_2.$$

Lemma B.0.5 (Gaussian mechanism [104]) *If a function $f : \mathcal{Z}^n \to \mathcal{R}^p$ has ℓ_2-sensitivity ϑ_2, then the randomized algorithm $\mathcal{M}$, which on input D outputs $f(D) + b$, where $b \sim \mathcal{N}(0, \sigma^2)^p$, satisfies (ϵ, δ)-DP, where $\sigma \geq c\vartheta_2/\epsilon$ and $c^2 > 2\log(1.25/\delta)$. The $\mathcal{N}(0, \sigma^2)^p$ denotes a vector of p i.i.d. samples from the Gaussian distribution $\mathcal{N}(0, \sigma^2)$.*

Algorithm 9 Sparse Vector Technique [162, 104]

Input: Dataset D, query set $\mathcal{Q} = \{q_1, ..., q_m\}$, privacy parameters $\epsilon, \delta > 0$, unstable query cutoff T, threshold w.

1: $c \leftarrow 0, \lambda \leftarrow \sqrt{32T\log(1/\delta)/\epsilon}, \hat{w} \leftarrow w + \texttt{Lap}(\lambda)$.
2: **for** $q \in \mathcal{Q}$ and $c \leq T$ **do**
3: $\quad \hat{q} \leftarrow q + \texttt{Lap}(2\lambda)$.
4: $\quad$ **if** $\hat{q} > \hat{w}$ **then**
5: $\quad\quad$ Output $\top$.
6: $\quad$ **else**
7: $\quad\quad$ Output $\perp$. $\hat{w} \leftarrow w + 1, c \leftarrow c + 1$.
8: $\quad$ **end if**
9: **end for**

Lemma B.0.6 (Privacy guarantee of Algorithm 9 [104]) *Algorithm 9 is (ϵ, δ)-DP.*

Lemma B.0.7 (Utility guarantee of Algorithm 9 [104]) *For*

$$\phi = \log(2mT/\beta)\sqrt{512T\log(1/\delta)}/\epsilon,$$

and any set of m queries, define the set $L(\phi) = \{i : q_i(D) \leq w + \phi\}$. If $|L(\phi)| \leq T$, then w.p. at least $1 - \beta : \forall i \notin L(\phi)$ Algorithm 9 outputs $\top$.

Definition B.0.8 (k-stability [114]) *A function $f : \mathcal{Z} \to \mathcal{R}$ is k stable on dataset D if adding or removing any k elements from D does not change the value of f, i.e., $f(D) = f(D')$ for all D' such that $|D - D'| \leq k$. We say f is stable on D if it is (at least) 1-stable on D, and unstable otherwise.*

Algorithm 10 Distance to Instability Framework [114]

Input: Dataset D, function $f : \mathcal{Z} \to \mathcal{R}$, distance to instability $\text{dist}_f : \mathcal{Z} \to \mathcal{R}$, thereshold Γ, privacy parameter $\epsilon > 0$.

1: $\widehat{\text{dist}} \leftarrow \widehat{\text{dist}}_f(D) + \texttt{Lap}(1/\epsilon)$.
2: **if** $\widehat{\text{dist}} > \Gamma$ **then**
3: Output $f(D)$.
4: **else**
5: Output $\perp$.
6: **end if**

Lemma B.0.9 (Privacy guarantee of Algorithm 10 [88]) *If the threshold $\Gamma = \log(1/\delta)/\epsilon$, and the distance to instability function $\text{dist}_f(D) = \text{argmax}_k(f(D)$ is k-stable), then Algorithm 10 is (ϵ, δ)-DP.*

Lemma B.0.10 (Utility guarantee of Algorithm 10 [114]) *If the threshold $\Gamma = \log(1/\delta)/\epsilon$, and the distance to instability function $\text{dist}_f(D) = \text{argmax}_k(f(D)$ is k-stable), and $f(D)$ is $((\log(1/\delta) + \log(1/\beta))/\epsilon)$-stable, then Algorithm 10 outputs $f(D)$ w.p. at least $1 - \beta$.*

Definition B.0.11 (Definition 1.1 of [105]) *$\mathcal{M}$ obeys (ξ, ρ)-zCDP if for two adjacent dataset D, D', for all $\phi \in (1, \infty)$, the Renyi-divergence of order ϕ below obeys that*

$$D_\phi(\mathcal{M}(D) \| \mathcal{M}(D')) \leq \xi + \rho\alpha.$$

When $\xi = 0$, we also call it ρ-zCDP (or simply ρ-CDP, since we are not considering other versions of CDPs in this thesis).

The following two lemmas will be used in the privacy analysis of the SVT-based PATE.

Lemma B.0.12 (Proposition 1.3 of [105]) *If $\mathcal{M}$ obeys ρ-zCDP, then $\mathcal{M}$ is $(\rho+2\sqrt{\rho \log(1/\delta)}, \delta)$-DP for any $\delta > 0$.*

Lemma B.0.13 (Proposition 1.4 of [105]) *If $\mathcal{M}$ obeys ϵ-DP, then $\mathcal{M}$ obeys $\frac{\epsilon^2}{2}$-CDP.*

Bibliography

[1] N. Nakamura, J. Seepaul, J. B. Kadane, and B. Reeja-Jayan, *Design for low-temperature microwave-assisted crystallization of ceramic thin films*, *Applied Stochastic Models in Business and Industry* **33** (2017), no. 3 314–321.

[2] A. E. Gongora, B. Xu, W. Perry, C. Okoye, P. Riley, K. G. Reyes, E. F. Morgan, and K. A. Brown, *A bayesian experimental autonomous researcher for mechanical design*, *Science Advances* **6** (2020), no. 15 1–6.

[3] C. Liu and Y.-X. Wang, *Global optimization with parametric function approximation*, in *International Conference on Machine Learning*, 2023.

[4] C. Liu, Y. Zhu, K. Chaudhuri, and Y.-X. Wang, *Revisiting model-agnostic private learning: Faster rates and active learning*, in *International Conference on Artificial Intelligence and Statistics*, 2021.

[5] C. Liu, Y. Zhu, K. Chaudhuri, and Y.-X. Wang, *Revisiting model-agnostic private learning: Faster rates and active learning*, *Journal of Machine Learning Research* **22** (2021), no. 262 1–44.

[6] N. Qiao, Y. Sun, C. Liu, L. Xia, J. Luo, K. Zhang, and C.-H. Kuo, *Human-in-the-loop video semantic segmentation auto-annotation*, in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

[7] C. Liu, A. R. Natarajan, D. E. Ober, A. Van der Ven, and Y.-X. Wang, *Cost-sensitive experimental design for atomistic modeling*, *ICML Workshop on Adaptive Experimental Design and Active Learning in the Real World* (2022).

[8] C. Liu and Y.-X. Wang, *Doubly robust crowdsourcing*, *Journal of Artificial Intelligence Research* **73** (2022) 209–229.

[9] C. Liu, P. Zhao, S.-J. Huang, Y. Jiang, and Z.-H. Zhou, *Dual set multi-label learning*, in *AAAI Conference on Artificial Intelligence*, 2018.

[10] C. Liu, M. Yin, and Y.-X. Wang, *No-regret linear bandits beyond realizability*, in *Conference on Uncertainty in Artificial Intelligence*, 2023.

[11] K. Kandasamy, K. R. Vysyaraju, W. Neiswanger, B. Paria, C. R. Collins, J. Schneider, B. Poczos, and E. P. Xing, *Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly*, *Journal of Machine Learning Research* **21** (2020), no. 81 1–27.

[12] P. I. Frazier, *A tutorial on bayesian optimization*, *arXiv preprint arXiv:1807.02811* (2018).

[13] P. Jain, P. Kar, *et. al.*, *Non-convex optimization for machine learning*, *Foundations and Trends® in Machine Learning* **10** (2017), no. 3-4 142–363.

[14] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, *Gaussian process optimization in the bandit setting: no regret and experimental design*, in *International Conference on Machine Learning*, 2010.

[15] R. D. Nowak, *Lecture notes: Complexity regularization for squared error loss*, 2007.

[16] Y. Abbasi-yadkori, D. Pál, and C. Szepesvári, *Improved algorithms for linear stochastic bandits*, in *Advances in Neural Information Processing Systems 24*, 2011.

[17] L. Li, Y. Lu, and D. Zhou, *Provably optimal algorithms for generalized linear contextual bandits*, in *International Conference on Machine Learning*, 2017.

[18] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun, *Reinforcement learning: Theory and algorithms*, 2021.

[19] A. Rinnooy Kan and G. T. Timmer, *Stochastic global optimization methods part i: Clustering methods*, *Mathematical programming* **39** (1987), no. 1 27–56.

[20] A. Rinnooy Kan and G. T. Timmer, *Stochastic global optimization methods part ii: Multi level methods*, *Mathematical Programming* **39** (1987), no. 1 57–78.

[21] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári, *X-armed bandits*, *Journal of Machine Learning Research* **12** (2011), no. 46 1655–1695.

[22] C. Malherbe and N. Vayatis, *Global optimization of lipschitz functions*, in *International Conference on Machine Learning*, 2017.

[23] E. Hazan, A. Klivans, and Y. Yuan, *Hyperparameter optimization: a spectral approach*, in *International Conference on Learning Representations*, 2018.

[24] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing, *Neural architecture search with bayesian optimisation and optimal transport*, in *Advances in neural information processing systems 31*, 2018.

[25] L. Wang, R. Fonseca, and Y. Tian, *Learning search space partition for black-box optimization using monte carlo tree search*, in *Advances in Neural Information Processing Systems 33*, 2020.

[26] P. I. Frazier and J. Wang, *Bayesian optimization for materials design*, in *Information science for materials discovery and design*, pp. 45–75. Springer, 2016.

[27] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, *Taking the human out of the loop: A review of bayesian optimization*, *Proceedings of the IEEE* **104** (2015), no. 1 148–175.

[28] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT Press, 2006.

[29] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient global optimization of expensive black-box functions*, *Journal of Global Optimization* **13** (1998), no. 4 455–492.

[30] A. D. Bull, *Convergence rates of efficient global optimization algorithms.*, *Journal of Machine Learning Research* **12** (2011), no. 10 2879–2904.

[31] P. Frazier, W. Powell, and S. Dayanik, *The knowledge-gradient policy for correlated normal beliefs*, *INFORMS Journal on Computing* **21** (2009), no. 4 599–613.

[32] S. Agrawal and N. Goyal, *Thompson sampling for contextual bandits with linear payoffs*, in *International Conference on Machine Learning*, 2013.

[33] X. Cai and J. Scarlett, *On lower bounds for standard and robust gaussian process bandit optimization*, in *International Conference on Machine Learning*, 2021.

[34] J. Scarlett, I. Bogunovic, and V. Cevher, *Lower bounds on regret for noisy gaussian process bandit optimization*, in *Annual Conference on Learning Theory*, 2017.

[35] S. Shekhar and T. Javidi, *Gaussian process bandits with adaptive discretization*, *Electronic Journal of Statistics* **12** (2018), no. 2 3829–3874.

[36] D. Calandriello, L. Carratino, A. Lazaric, M. Valko, and L. Rosasco, *Gaussian process optimization with adaptive sketching: Scalable and no regret*, in *Annual Conference on Learning Theory*, 2019.

[37] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, *Scalable global optimization via local bayesian optimization*, in *Advances in Neural Information Processing Systems 32*, 2019.

[38] S. Salgia, S. Vakili, and Q. Zhao, *A domain-shrinking based bayesian optimization algorithm with order-optimal regret performance*, in *Advances in Neural Information Processing Systems 34*, 2021.

[39] M. Rando, L. Carratino, S. Villa, and L. Rosasco, *Ada-bkb: Scalable gaussian process optimization on continuous domains by adaptive discretization*, in *International Conference on Artificial Intelligence and Statistics*, 2022.

[40] Y. Wang, S. Balakrishnan, and A. Singh, *Optimization of smooth functions with noisy observations: Local minimax rates*, in *Advances in Neural Information Processing Systems 31*, 2018.

[41] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams, *Scalable bayesian optimization using deep neural networks*, in *International Conference on Machine Learning*, 2015.

[42] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, *Bayesian optimization with robust bayesian neural networks*, in *Advances in Neural Information Processing Systems 29*, 2016.

[43] Y. Li, Y. Wang, and Y. Zhou, *Nearly minimax-optimal regret for linearly parameterized bandits*, in *Annual Conference on Learning Theory*, 2019.

[44] D. Foster and A. Rakhlin, *Beyond ucb: Optimal and efficient contextual bandits with regression oracles*, in *International Conference on Machine Learning*, 2020.

[45] D. Russo and B. Van Roy, *Eluder dimension and the sample complexity of optimistic exploration*, in *Advances in Neural Information Processing Systems 26*, 2013.

[46] S. Filippi, O. Cappe, A. Garivier, and C. Szepesvári, *Parametric bandits: The generalized linear case*, in *Advances in Neural Information Processing Systems 23*, 2010.

[47] D. Zhou, L. Li, and Q. Gu, *Neural contextual bandits with ucb-based exploration*, in *International Conference on Machine Learning*, 2020.

[48] W. Zhang, D. Zhou, L. Li, and Q. Gu, *Neural thompson sampling*, in *International Conference on Learning Representations*, 2020.

[49] Z. Dai, Y. Shu, B. K. H. Low, and P. Jaillet, *Sample-then-optimize batch neural Thompson sampling*, in *Advances in Neural Information Processing Systems 35*, 2022.

[50] W. Chu, L. Li, L. Reyzin, and R. Schapire, *Contextual bandits with linear payoff functions*, in *International Conference on Artificial Intelligence and Statistics*, 2011.

[51] D. Foster, A. Agarwal, M. Dudik, H. Luo, and R. Schapire, *Practical contextual bandits with regression oracles*, in *International Conference on Machine Learning*, 2018.

[52] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[53] L. Zhang, T. Yang, J. Yi, R. Jin, and Z.-H. Zhou, *Improved dynamic regret for non-degenerate functions*, in *Advances in Neural Information Processing Systems 30*, 2017.

[54] V. Dani, T. P. Hayes, and S. M. Kakade, *Stochastic linear optimization under bandit feedback*, in *Annual Conference on Learning Theory*, 2008.

[55] H. J. Kushner, *A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise*, *Journal of Basic Engineering* **8** (1964), no. 1 97–106.

[56] M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, *Botorch: a framework for efficient monte-carlo bayesian optimization*, in *Advances in Neural Information Processing Systems 33*, 2020.

[57] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi, "scikit-optimize." https://scikit-optimize.github.io, 2021.

[58] D. Dua and C. Graff, *UCI machine learning repository*, 2017.

[59] E. Claeys, P. Gancarski, M. Maumy-Bertrand, and H. Wassner, *Dynamic allocation optimization in a/b-tests using classification-based preprocessing*, *IEEE Transactions on Knowledge and Data Engineering* **35** (2021), no. 1 335–349.

[60] S. Wang, Q. Liu, T. Ge, D. Lian, and Z. Zhang, *A hybrid bandit model with visual priors for creative ranking in display advertising*, in *The Web Conference*, 2021.

[61] A. Moradipari, C. Thrampoulidis, and M. Alizadeh, *Stage-wise conservative linear bandits*, in *Advances in Neural Information Processing Systems 33*, 2020.

[62] A. Alieva, A. Cutkosky, and A. Das, *Robust pure exploration in linear bandits with limited budget*, in *International Conference on Machine Learning*, 2021.

[63] J. Katz-Samuels, L. Jain, K. G. Jamieson, *et. al.*, *An empirical process approach to the union bound: Practical algorithms for combinatorial and linear bandits*, in *Advances in Neural Information Processing Systems 33*, 2020.

[64] A. Ghosh, S. R. Chowdhury, and A. Gopalan, *Misspecified linear bandits*, in *AAAI Conference on Artificial Intelligence*, 2017.

[65] T. Lattimore, C. Szepesvari, and G. Weisz, *Learning with good feature representations in bandits and in rl with a generative model*, in *International Conference on Machine Learning*, 2020.

[66] A. Zanette, A. Lazaric, M. Kochenderfer, and E. Brunskill, *Learning near optimal policies with low inherent bellman error*, in *International Conference on Machine Learning*, 2020.

[67] G. Neu and J. Olkhovskaya, *Efficient and robust algorithms for adversarial linear contextual bandits*, in *Annual Conference on Learning Theory*, 2020.

[68] I. Bogunovic and A. Krause, *Misspecified gaussian process bandit optimization*, in *Advances in Neural Information Processing Systems 34*, 2021.

[69] S. K. Krishnamurthy, V. Hadad, and S. Athey, *Tractable contextual bandits beyond realizability*, in *International Conference on Artificial Intelligence and Statistics*, 2021.

[70] N. Abe and P. M. Long, *Associative reinforcement learning using linear probabilistic concepts*, in *International Conference on Machine Learning*, 1999.

[71] P. Auer, N. Cesa-Bianchi, and P. Fischer, *Finite-time analysis of the multiarmed bandit problem*, *Machine learning* **47** (2002) 235–256.

[72] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[73] D. J. Foster, C. Gentile, M. Mohri, and J. Zimmert, *Adapting to misspecification in contextual bandits*, in *Advances in Neural Information Processing Systems 33*, 2020.

[74] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire, *Taming the monster: A fast and simple algorithm for contextual bandits*, in *International Conference on Machine Learning*, 2014.

[75] C. Dwork, F. McSherry, K. Nissim, and A. Smith, *Calibrating noise to sensitivity in private data analysis*, in *Theory of Cryptography Conference*, 2006.

[76] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, *Privacy: Theory meets practice on the map*, in *International Conference on Data Engineering*, pp. 277–286, 2008.

[77] Ú. Erlingsson, V. Pihur, and A. Korolova, *Rappor: Randomized aggregatable privacy-preserving ordinal response*, in *ACM Conference on Computer and Communications Security*, 2014.

[78] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, *Learning differentially private recurrent language models*, in *International Conference on Learning Representations*, 2018.

[79] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, *What can we learn privately?*, *SIAM Journal on Computing* **40** (2011), no. 3 793–826.

[80] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, *Differentially private empirical risk minimization*, *Journal of Machine Learning Research* **12** (2011), no. 3 1069–1109.

[81] R. Bassily, A. Smith, and A. Thakurta, *Private empirical risk minimization: Efficient algorithms and tight error bounds*, in *IEEE Symposium on Foundations of Computer Science*, 2014.

[82] Y.-X. Wang, S. Fienberg, and A. Smola, *Privacy for free: Posterior sampling and stochastic gradient monte carlo*, in *International Conference on Machine Learning*, 2015.

[83] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, *Deep learning with differential privacy*, in *ACM Conference on Computer and Communications Security*, 2016.

[84] R. Shokri and V. Shmatikov, *Privacy-preserving deep learning*, in *ACM Conference on Computer and Communications Security*, 2015.

[85] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, *Semi-supervised knowledge transfer for deep learning from private training data*, in *International Conference on Learning Representations*, 2017.

[86] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, *Scalable private learning with pate*, in *International Conference on Learning Representations*, 2018.

[87] K. Nissim, S. Raskhodnikova, and A. Smith, *Smooth sensitivity and sampling in private data analysis*, in *ACM Symposium on Theory of Computing*, 2007.

[88] R. Bassily, O. Thakkar, and A. G. Thakurta, *Model-agnostic private learning*, in *Advances in Neural Information Processing Systems 31*, 2018.

[89] M. Bun, K. Nissim, U. Stemmer, and S. Vadhan, *Differentially private release and learning of threshold functions*, in *IEEE Symposium on Foundations of Computer Science*, 2015.

[90] Y.-X. Wang, J. Lei, and S. E. Fienberg, *Learning with differential privacy: Stability, learnability and the sufficiency and necessity of erm principle*, *Journal of Machine Learning Research* **17** (2016), no. 183 1–40.

[91] E. Mammen and A. B. Tsybakov, *Smooth discrimination analysis*, *Annals of Statistics* **27** (1999), no. 6 1808–1829.

[92] A. B. Tsybakov, *Optimal aggregation of classifiers in statistical learning*, *Annals of Statistics* **32** (2004), no. 1 135–166.

[93] S. Hanneke *et. al.*, *Theory of disagreement-based active learning*, *Foundations and Trends® in Machine Learning* **7** (2014), no. 2-3 131–309.

[94] Z. Zhao, N. Papernot, S. Singh, N. Polyzotis, and A. Odena, *Improving differentially private models with active learning*, *arXiv preprint arXiv:1910.01177* (2019).

[95] K. Chaudhuri and D. Hsu, *Sample complexity bounds for differentially private learning*, in *Annual Conference on Learning Theory*, 2011.

[96] N. Alon, R. Bassily, and S. Moran, *Limits of private learning with access to public data*, in *Advances in Neural Information Processing Systems 32*, 2019.

[97] A. Beimel, K. Nissim, and U. Stemmer, *Private learning and sanitization: Pure vs. approximate differential privacy*, *Theory of Computing* **12** (2016), no. 890 1–61.

[98] C. Dwork and V. Feldman, *Privacy-preserving prediction*, in *Annual Conference on Learning Theory*, 2018.

[99] Y. Dagan and V. Feldman, *Pac learning with stable and private predictions*, in *Annual Conference on Learning Theory (COLT-20)*, pp. 1389–1410, 2020.

[100] A. Nandi and R. Bassily, *Privately answering classification queries in the agnostic pac model*, in *International Conference on Algorithmic Learning Theory (ALT-20)*, pp. 687–703, 2020.

[101] A. Beimel, K. Nissim, and U. Stemmer, *Characterizing the sample complexity of private learners*, in *Innovations in Theoretical Computer Science Conference*, 2013.

[102] S. Boucheron, O. Bousquet, and G. Lugosi, *Theory of classification: A survey of some recent advances*, *ESAIM: Probability and Statistics* **9** (2005) 323–375.

[103] C. Zhang and K. Chaudhuri, *Beyond disagreement-based agnostic active learning*, in *Advances in Neural Information Processing Systems 27*, 2014.

[104] C. Dwork and A. Roth, *The algorithmic foundations of differential privacy*, *Foundations and Trends in Theoretical Computer Science* **9** (2014), no. 3–4 211–407.

[105] M. Bun and T. Steinke, *Concentrated differential privacy: Simplifications, extensions, and lower bounds*, in *Theory of Cryptography Conference*, 2016.

[106] R. Bassily, O. Thakkar, and A. Thakurta, *Model-agnostic private learning via stability*, *arXiv preprint arXiv:1803.05101* (2018).

[107] O. Bousquet, S. Boucheron, and G. Lugosi, *Introduction to statistical learning theory*, *Advanced Lectures on Machine Learning: ML Summer Schools* (2004) 169–207.

[108] V. N. Vapnik, *The nature of statistical learning theory*. Springer, 1995.

[109] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, *Learnability, stability and uniform convergence*, *Journal of Machine Learning Research* **11** (2010), no. 90 2635–2670.

[110] S. Yan, K. Chaudhuri, and T. Javidi, *Active learning with logged data*, in *International Conference on Machine Learning*, 2018.

[111] Y.-X. Wang, B. Balle, and S. Kasiviswanathan, *Subsampled rényi differential privacy and analytical moments accountant*, in *International Conference on Artificial Intelligence and Statistics*, 2019.

[112] B. Balle and Y.-X. Wang, *Improving gaussian mechanism for differential privacy: Analytical calibration and optimal denoising*, in *International Conference in Machine Learning (ICML-18)*, 2018.

[113] M. Hardt and G. N. Rothblum, *A multiplicative weights mechanism for privacy-preserving data analysis*, in *IEEE Symposium on Foundations of Computer Science*, 2010.

[114] A. G. Thakurta and A. Smith, *Differentially private feature selection via stability arguments, and the robustness of the lasso*, in *Annual Conference on Learning Theory*, 2013.

[115] A. Geiger, P. Lenz, and R. Urtasun, *Are we ready for autonomous driving? the kitti vision benchmark suite*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2012.

[116] B.-C.-Z. Blaga and S. Nedevschi, *Semantic segmentation learning for autonomous uavs using simulators and real data*, in *Conference on Intelligent Computer Communication and Processing*, 2019.

[117] T.-y. Ko and S.-h. Lee, *Novel method of semantic segmentation applicable to augmented reality*, *Sensors* **20** (2020), no. 6 1737.

[118] W. Ren, J. Pan, X. Cao, and M.-H. Yang, *Video deblurring via semantic segmentation and pixel-wise non-linear kernel*, in *IEEE/CVF International Conference on Computer Vision*, 2017.

[119] W. Ren, J. Zhang, X. Xu, L. Ma, X. Cao, G. Meng, and W. Liu, *Deep video dehazing with semantic segmentation*, *IEEE Transactions on Image Processing* **28** (2019), no. 4 1895–1908.

[120] J. Ji, S. Buch, A. Soto, and J. C. Niebles, *End-to-end joint semantic segmentation of actors and actions in video*, in *European Conference on Computer Vision*, 2018.

[121] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, *Joint semantic segmentation and 3d reconstruction from monocular video*, in *European Conference on Computer Vision*, 2014.

[122] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, *The cityscapes dataset for semantic urban scene understanding*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[123] G. J. Brostow, J. Fauqueur, and R. Cipolla, *Semantic object classes in video: A high-definition ground truth database*, *Pattern Recognition Letters* **30** (2009), no. 2 88–97.

[124] I. Budvytis, V. Badrinarayanan, and R. Cipolla, *Label propagation in complex video sequences using semi-supervised learning.*, in *British Machine Vision Conference*, 2010.

[125] V. Badrinarayanan, F. Galasso, and R. Cipolla, *Label propagation in video sequences*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2010.

[126] I. Budvytis, P. Sauer, T. Roddick, K. Breen, and R. Cipolla, *Large scale labelled video data augmentation for semantic segmentation in driving scenarios*, in *International Conference on Computer Vision Workshops*, 2017.

[127] V. Badrinarayanan, I. Budvytis, and R. Cipolla, *Semi-supervised video segmentation using tree structured graphical models*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35** (2013), no. 11 2751–2764.

[128] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, *Suggestive annotation: A deep active learning framework for biomedical image segmentation*, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2017.

[129] S. Sinha, S. Ebrahimi, and T. Darrell, *Variational adversarial active learning*, in *IEE/CVF International Conference on Computer Vision*, 2019.

[130] C. Dai, S. Wang, Y. Mo, K. Zhou, E. Angelini, Y. Guo, and W. Bai, *Suggestive annotation of brain tumour images with gradient-guided sampling*, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2020.

[131] R. Mackowiak, P. Lenz, O. Ghori, F. Diego, O. Lange, and C. Rother, *Cereals - cost-effective region-based active learning for semantic segmentation*, in *British Machine Vision Conference*, 2018.

[132] A. Casanova, P. O. Pinheiro, N. Rostamzadeh, and C. J. Pal, *Reinforced active learning for image segmentation*, in *International Conference on Learning Representations*, 2020.

[133] P. Colling, L. Roese-Koerner, H. Gottschalk, and M. Rottmann, *Metabox+: a new region based active learning method for semantic segmentation using priority maps*, in *International Conference on Pattern Recognition Applications and Methods*, 2021.

[134] Y. Siddiqui, J. Valentin, and M. Nießner, *Viewal: Active learning with viewpoint entropy for semantic segmentation*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[135] L. Cai, X. Xu, J. H. Liew, and C. S. Foo, *Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[136] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun, *Predicting deeper into the future of semantic segmentation*, in *IEEE/CVF International Conference on Computer Vision*, 2017.

[137] S. K. Mustikovela, M. Y. Yang, and C. Rother, *Can ground truth label propagation from video help semantic segmentation?*, in *European Conference on Computer Vision*, 2016.

[138] R. Gadde, V. Jampani, and P. V. Gehler, *Semantic video cnns through representation warping*, in *IEEE/CVF International Conference on Computer Vision*, 2017.

[139] D. Nilsson and C. Sminchisescu, *Semantic video segmentation by gated recurrent flow propagation*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[140] B. Settles, *Active learning literature survey*, tech. rep., University of Wisconsin–Madison, 2009.

[141] A. Fathi, M. F. Balcan, X. Ren, and J. M. Rehg, *Combining self training and active learning for video segmentation*, in *British Machine Vision Conference*, 2011.

[142] S. Vijayanarasimhan and K. Grauman, *Active frame selection for label propagation in videos*, in *European Conference on Computer Vision*, 2012.

[143] A. Abad, M. Nabi, and A. Moschitti, *Autonomous crowdsourcing through human-machine collaborative learning*, in *ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017.

[144] M. Ravanbakhsh, T. Klein, K. Batmanghelich, and M. Nabi, *Uncertainty-driven semantic segmentation through human-machine collaborative learning*, in *Medical Imaging with Deep Learning*, 2019.

[145] Y. Heo, Y. Jun Koh, and C.-S. Kim, *Interactive video object segmentation using global and local transfer modules*, in *European Conference on Computer Vision*, 2020.

[146] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, *Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[147] J.-Y. Huang and J.-J. Ding, *Generic image segmentation in fully convolutional networks by superpixel merging map*, in *Asian Conference on Computer Vision*, 2020.

[148] Z. Teed and J. Deng, *Raft: Recurrent all-pairs field transforms for optical flow*, in *European Conference on Computer Vision*, 2020.

[149] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, *Improving semantic segmentation via video propagation and label relaxation*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[150] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et. al.*, *Deep high-resolution representation learning for visual recognition*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43** (2021), no. 10 3349–3364.

[151] S. Ruder, *An overview of gradient descent optimization algorithms*, *arXiv:1609.04747* (2016).

[152] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, *Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation?*, in *IEEE/CVF International Conference on Computer Vision*, 2017.

[153] F. Sadat Saleh, M. Sadegh Aliakbarian, M. Salzmann, L. Petersson, and J. M. Alvarez, *Effective use of synthetic data for urban scene semantic segmentation*, in *European Conference on Computer Vision*, 2018.

[154] G. Bradski, *The OpenCV Library*, *Dr. Dobb's Journal of Software Tools* (2000).

[155] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *Pytorch: An imperative style, high-performance deep learning library*, in *Advances in Neural Information Processing Systems 32*, 2019.

[156] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov, *Boundary IoU: Improving object-centric image segmentation evaluation*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[157] Y. Liu, C. Shen, C. Yu, and J. Wang, *Efficient semantic video segmentation with per-frame inference*, in *European Conference on Computer Vision*, 2020.

[158] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, *Scannet: Richly-annotated 3d reconstructions of indoor scenes*, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[159] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, *Indoor segmentation and support inference from rgbd images*, in *European Conference on Computer Vision*, 2012.

[160] J. González, Z. Dai, A. Damianou, and N. D. Lawrence, *Preferential bayesian optimization*, in *International Conference on Machine Learning*, pp. 1282–1291, 2017.

[161] J. Sherman and W. J. Morrison, *Adjustment of an inverse matrix corresponding to a change in one element of a given matrix*, *Annals of Mathematical Statistics* **21** (1950), no. 1 124–127.

[162] C. Dwork, G. N. Rothblum, and S. Vadhan, *Boosting and differential privacy*, in *IEEE Symposium on Foundations of Computer Science*, 2010.