Optimal Query Complexity of Secure Stochastic Convex Optimization

Wei Tang[†]**, Chien-Ju Ho**[†]**, and Yang Liu**[*]
[†]Washington University in St. Louis, [*]UC Santa Cruz
{w.tang, chienju.ho}@wustl.edu, yangliu@ucsc.edu

Abstract

We study the *secure* stochastic convex optimization problem. A learner aims to learn the optimal point of a convex function through sequentially querying a (stochastic) gradient oracle. In the meantime, there exists an adversary who aims to free-ride and infer the learning outcome of the learner from observing the learner's queries. The adversary observes only the points of the queries but not the feedback from the oracle. The goal of the learner is to optimize the accuracy, i.e., obtaining an accurate estimate of the optimal point, while securing her privacy, i.e., making it difficult for the adversary to infer the optimal point. We formally quantify this tradeoff between learner's accuracy and privacy and characterize the lower and upper bounds on the learner's query complexity as a function of desired levels of accuracy and privacy. For the analysis of lower bounds, we provide a general template based on information theoretical analysis and then tailor the template to several families of problems, including stochastic convex optimization and (noisy) binary search. We also present a generic secure learning protocol that achieves the matching upper bound up to logarithmic factors.

1 Introduction

Optimization, that seeks to find the optimal point of a function, is an important tool in various domains, including decision making and machine learning. Modern optimization techniques, such as gradient descent, often run in an iterative manner: the learner adaptively queries a (noisy) oracle, obtains the information about the function (e.g., gradient) at the query point, and updates the estimate of the optimal point. While such iterative techniques have been well studied and shown to be efficient, the iterative nature introduces potential risks of information leak. A spying adversary, who can observe the series of query points the learner sends to the oracle but not the oracle responses, may free-ride and infer the optimal point from the queries alone.

For example, consider a company aiming to find the optimal price for a new product. The company might hire market research firm that performs dynamic pricing on a test population. Assume the market research firm is adopting an optimization algorithm that increases the price if the sale happens and decreases the price otherwise. An adversary (e.g., a competing company), who knows the algorithm and can observe the price changes (e.g., by entering the test population), may infer and estimate the optimal price before the product launch even without knowing whether the transaction happens or not during market research. As another example, in federated learning, the learner might aim to optimize the parameters of their learning models using gradient decent. Since data might be distributed, the learner needs to sequentially broadcast their models to data-holding users in order to obtain the gradient information. An adversary can pretend to be data-holding user to receive the

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

sequence of broadcasted models. He might then estimate the final model even without obtaining the gradient information.

In this work, we study the *secure* stochastic convex optimization problem, in which the learner aims to optimize the *accuracy*, i.e., obtain an accurate estimate to the optimal point, while securing her *privacy*, i.e., preventing an adversary from inferring what she learned[1]. We formalize the notions of accuracy and privacy using PAC (Probably Approximate Correct) style notions. The algorithm is (ϵ, δ)-accurate if the learner's estimate is within ϵ distance to the optima with probability at least $1-\delta$. The algorithm is $(\epsilon^{\mathrm{adv}}, \delta^{\mathrm{adv}})$-private[2] if for any adversary that can infer from only the query points, the probability for his estimate to be within ϵ^{adv} distance to the optima is at most δ^{adv}. Our goal is to characterize the trade-offs between learner's accuracy and privacy using query complexity, i.e., the minimum number of queries needed to achieve a given level of accuracy and privacy.

Our main results include the characterization of the lower and upper bounds of the query complexity for the secure stochastic convex optimization problem. In particular, we study the general κ-uniformly convex functions. We show that, with logarithmic factors compressed in the bounds, when the error measure is function error (i.e., the error is the difference of the objective function values between the estimate and the optima), we obtain matching upper and lower bounds in the order of $\Theta\left(1/(\delta^{\mathrm{adv}}\epsilon^{(2\kappa-2)/\kappa})\right)$. When the error measure is point error (i.e., the error is the difference between the estimate and optima in the input domain), we obtain matching upper and lower bounds in the order of $\Theta\left(1/(\delta^{\mathrm{adv}}\epsilon^{2\kappa-2})\right)$. Our results recover the classic complexity bounds in convex optimization (strongly convex for $\kappa = 2$ and convex for $\kappa \to \infty$) when there is no requirement to secure the learner's privacy. Our bounds suffer an additional factor of $\Theta(1/\delta^{\mathrm{adv}})$ compared to classic non-secure bounds[3], which can be viewed as a complexity price that the learner has to pay to secure her privacy.

To highlight our technical contributions, for the lower-bound analysis, we develop a general template based on an information-theoretical analysis for convex programming [13]. In addition to deriving the lower bound, we demonstrate that the same template can be applied to obtain the same lower bound of private binary search [23], in which the authors focus on a (Bayesian) binary search problem and assume the learner has a uniform prior on where the target is and has access to a *noiseless* oracle. In addition to obtaining the same lower bound using different techniques, we show that the template offers the lower bound for private *noisy* binary search, which has been also discussed in a recent work [21]. As for the upper bound, we propose a secure learning protocol that is immune to any adversary. The protocol may incorporate an arbitrary non-secure but efficient learning algorithm as a subroutine, and a matching upper bound up to logarithmic factors is proved.

Related work. This paper is closely related to the recent works in private sequential learning [23, 19, 21], which study private Bayesian binary search: A learner aims to estimate an unknown target value through sequentially querying an oracle which returns exact binary responses, while protecting her estimations from an adversary. The authors assume that the learner has a uniform prior for the unknown target value. We generalize their setting of binary search to stochastic convex optimization and adopts different analysis which builds on minimax bounds instead of assuming uniform prior.

Another close line of research is differentially private online learning [2, 4, 6, 7, 10, 17, 18]. Our work departs significantly from these works. In differential privacy, the goal is to ensure the change for any individual participant does not change the outcome substantially, and therefore the privacy of individuals is protected. The goal of our work is to secure the learner's privacy in the sense that the adversary cannot infer what the learner is learning from observing the actions of the learner. We name our work *secure* optimization (where the learner's objective is secured from the adversary) to

[1] In this paper, we use "she" to address the learner and "he" to address the adversary. In addition, we denote our problem as *secure* optimization instead of *private* optimization to differentiate with the works in differential privacy. Generally speaking, the goal of differential privacy is to protect the privacy of individual data contributors, while our goal is to secure the privacy of the learner.

[2] We use superscript adv for the privacy notion since it is related to the adversary's estimation.

[3] The dependency on ϵ^{adv} is in the logarithmic factor.

emphasize this difference. Our technique is built on the minimax analysis for (stochastic) convex optimization problem [1, 5, 9, 11–13, 15, 16]. Our results complement this line of work through incorporating the privacy requirement.

2 Problem Formulation

Consider a *learner* $\mathcal{A}$ who aims to maximize the *accuracy* of learning the optimal point of an unknown convex function f through sequentially querying an oracle ϕ about the function information. In the meantime, the learner wants to secure her *privacy*, i.e., preventing a spying *adversary* from free-riding and inferring the learning outcome through observing where the learner queries. A problem class of convex optimization problem is defined by a triple $\mathcal{P} = (\mathcal{X}, \mathcal{F}, \phi)$, where $\mathcal{X} \subset \mathbb{R}^d$ is a compact and convex problem domain, $\mathcal{F}$ is a class of convex functions, and for any function $f \in \mathcal{F}$, $\phi : \mathcal{X} \times f \to \mathcal{Y}$ is an oracle function that answers any query $x \in \mathcal{X}$ by returning an element $\phi(x, f)$ in an information set $\mathcal{Y}$.

At the beginning of the learning process, an unknown convex objective function f is drawn from $\mathcal{F}$. Let x_f^* be the minimizer of f, i.e., $x_f^* = \arg\min_{x \in \mathcal{X}} f(x)$, and $f^* = f(x_f^*)$ be the optimal function value. At each time $t = 1$ to T, the learner submits a query X_t to the oracle and obtains a response $Y_t = \phi(X_t, f)$. Let $X^T = \{X_1, \ldots, X_T\}$ denote the set of queries till time T. Similarly $Y^T = \{Y_1, \ldots, Y_T\}$ denotes the set of corresponding responses. The learner can observe all queries and responses, i.e., X^T and Y^T, while the adversary can only observe the queries X^T. At the end of the learning, the learner outputs an estimate $\widehat{X}$ for x_f^*, based on X^T and Y^T, while the adversary outputs another estimate $\widehat{X}^{\text{adv}}$ based only on the query points X^T but not the responses.

Objective. The learner aims to design an algorithm $\mathcal{A}$, which sequentially decides X_t and formulates a candidate optimizer $\widehat{X}$ (optimizer here is equivalent to the estimate), with the goal of minimizing the number of queries while ensuring *accuracy*, i.e., $\widehat{X}$ is a good estimate to the optimal point x_f^*, and securing *privacy*, i.e., $\widehat{X}^{\text{adv}}$ is sufficiently far away from x_f^* for any adversary.

We use $\texttt{err}(\widehat{X}, f)$ to measure how close an estimate $\widehat{X}$ is to the optimal point of function f. Two generic error measures are: function error $\texttt{err}(\widehat{X}, f) = |f(\widehat{X}) - f^*|$ and point error $\texttt{err}(\widehat{X}, f) = \|\widehat{X} - x_f^*\|$, where $\|\cdot\|$ denotes the Euclidean norm. With the error measure in place, we formally define the notions of learner's accuracy and privacy requirements:

Definition 1 ((ϵ, δ)-accurate)**.** Fix $\epsilon, \delta \in (0, 1)$. Given a problem class $\mathcal{P} = (\mathcal{X}, \mathcal{F}, \phi)$, a learner's algorithm $\mathcal{A}$ is (ϵ, δ)-accurate if for any $f \in \mathcal{F}$,

$$\mathbb{P}(\texttt{err}(\widehat{X}, f) \geq \epsilon) \leq \delta, \tag{1}$$

where the probability is measured with respect to the randomness in the oracle's responses and the possible randomness in the algorithm.

We restrict the discussion to a general class of *reasonable* adversaries. In the following discussion, we say an adversary is reasonable if he is oblivious and *consistent*. In particular, an adversary is oblivious if he determines the estimation strategy ahead of the game. This oblivious assumption is commonly made in online learning literature [3, 8]. We also assume that the adversary is *consistent* as stated below. First, we assume the adversary has uniform *prior* beliefs about the optimizer. Upon observing information, the adversary updates his belief on where the optimizer is. The updated beliefs must be consistent with the prior in the sense that the expected updated beliefs over the randomness of the queries are the same as the prior. Formally, let $F(\cdot|\text{queries})$ denote the adversary's belief of the optimizer given the observed queries, then $\mathbb{E}[F(\cdot|\text{queries})] = \text{Prior}$, where the expectation is over the queries. If the adversary has equal beliefs on a set of estimates which may be the optimizer, he will generate the estimate uniformly at random among them.

Definition 2 (($\epsilon^{\text{adv}}, \delta^{\text{adv}}$)-private)**.** Fix $\epsilon^{\text{adv}}, \delta^{\text{adv}} \in (0,1)$. A learner's algorithm is $(\epsilon^{\text{adv}}, \delta^{\text{adv}})$-private if, for any estimator $\widehat{X}^{\text{adv}}$ generated by a reasonable adversary [4] and for any $f \in \mathcal{F}$,

$$\mathbb{P}(\texttt{err}(\widehat{X}^{\text{adv}}, f) \leq \epsilon^{\text{adv}}) \leq \delta^{\text{adv}}, \tag{2}$$

where the probability is measured with respect to the randomness in the oracle's responses, the algorithm, and the adversary estimator.

Remark 1. We choose to use the term "private" here in the definition in the sense that the algorithm aims to secure the privacy of the learner.

Intuitively, an algorithm is (ϵ, δ)-accurate if the estimate $\widehat{X}$ is within ϵ distance to the optima with probability at least $1-\delta$, and an algorithm is $(\epsilon^{\text{adv}}, \delta^{\text{adv}})$-private if for any adversary, with probability at most δ^{adv}, the estimate $\widehat{X}^{\text{adv}}$ is within ϵ^{adv} to the optima.

The goal of the learner is to minimize the number of queries while satisfying the requirements of achieving a given level of accuracy and securing her privacy. To characterize this goal, we define *secure query complexity* as follows:

Definition 3 (Secure Query Complexity)**.** Given a problem class $\mathcal{P} = (\mathcal{X}, \mathcal{F}, \phi)$, the secure query complexity $T_{\mathcal{P}}(\epsilon, \delta, \epsilon^{\text{adv}}, \delta^{\text{adv}})$ is defined as the least number of queries needed for a learner's algorithm to be simultaneously (ϵ, δ)-accurate and $(\epsilon^{\text{adv}}, \delta^{\text{adv}})$-private for any function $f \in \mathcal{F}$.

When it is clear from the context, we drop the input parameters and simply write $T_{\mathcal{P}}$.

2.1 Problem Classes

We illustrate the problem classes $\mathcal{P} = (\mathcal{X}, \mathcal{F}, \phi)$ that we explore in this work.

Types of oracle. We focus on settings in which the oracle returns the first-order information (as is common in gradient-based optimization algorithms). In particular, let $g(x)$ be an arbitrary subgradient in $\partial f(x)$. If the oracle only returns the sign of $g(x)$, we denote such oracle by ϕ^{sign}. A noisy sign oracle with correct probability being $p \in (0.5, 1)$ will be denoted by $\phi^{\mathsf{sign},p}$. We also consider the standard noisy first-order oracle that returns noisy 0-th and 1-st order information, where the information consists of the pair $(f(x)+Z_1, g(x)+Z_2)$, with the noise Z_1 added to the function value being drawn from $\mathcal{N}(0, \sigma^2)$ (zero-mean Gaussian distribution) and the noise Z_2 to the first-order information being drawn from $\mathcal{N}(0, \sigma^2 \mathcal{I}_d)$. We use $\phi^{(1)}$ to denote such noisy first-order oracle and refer it as the Gaussian oracle.

(Noisy) binary search. One of the simplest setups of our framework is the one-dimensional binary search, in which $\mathcal{X} = [0,1]$, $\mathcal{F}^{\text{Abs}} = \left\{ f(x) \triangleq |x - x^*| \right\}$, and the oracle is ϕ^{sign} (i.e., whether the query x is larger than the optimal x^*). The above setting can extend to a noisy binary search with oracle $\phi^{\mathsf{sign},p}$.

Convex optimization. We also explore the general convex optimization problem with first-order oracle $\phi^{(1)}$. We consider the general class of κ-uniformly convex function. Given $\kappa \geq 2$, let $\mathcal{F}^{\kappa}$ be the set of all convex functions that satisfy: $f(x) - f(x_f^*) \geq \frac{\lambda}{2}\|x - x_f^*\|^{\kappa}, \forall x \in \mathcal{X}$, for some $\lambda > 0$. κ-uniformly convex function is a general representation of convex functions: when $\kappa = 2$, it recovers strong convexity, and when $\kappa \to \infty$, it recovers (non-strong) convexity. We shall always assume the functions in $\mathcal{F}^{\kappa}$ are L-Lipschitz, i.e., for all $f \in \mathcal{F}^{\kappa}$ and all $x, y \in \mathcal{X}$, $\|f(x) - f(y)\| \leq L\|x - y\|$.

3 Lower Bounds on Secure Query Complexity

In this section, we characterize the hardness of our secure convex optimization problem by proving the lower bounds for secure query complexity $T_{\mathcal{P}}(\epsilon, \delta, \epsilon^{\text{adv}}, \delta^{\text{adv}})$. We first present a general approach

[4] Our results and analysis require the adversary to be *reasonable*, i.e., oblivious and consistent. The NeurIPS 2020 version does not spell out the assumption explicitly. We thank Jiaming Xu, Kuang Xu, and Dana Yang [22] for pointing this out.

for characterizing the lower bounds which may hold for most problem classes, with the results summarized in Theorem 1. We then demonstrate how to utilize this general approach to derive lower bounds for a variety of classes of problems in Section 3.2.

3.1 A general framework for characterizing lower bounds

Without the requirement to secure the learner's privacy, characterizing the query complexity can follow the proof techniques developed in minimax bounds literature via reducing the optimization problem into a hypothesis testing one [25, 24]. On a high-level, we can first construct a difficult problem subclass with a set of hard-to-differentiate functions. If there exists an optimization algorithm that achieves high accuracy, we can utilize the algorithm to differentiate functions in the set. Since there exist information bounds in hypothesis testing to characterize the hardness of differentiating functions, these information bounds imply the hardness of designing optimization algorithms that achieves high accuracy.

The main challenge we face is to incorporate the requirement of securing the learner's privacy in the analysis. Recall that the secure query complexity is defined with respect to all possible adversaries, and a stronger adversary makes it harder to maintain privacy. In our proof, we focus on an ostensibly weak adversary and derive our lower bounds with respect to this adversary. While this choice seems to lead to a weaker lower bound, we demonstrate later that there is a matching upper bound for any adversary. These two results jointly imply that no other adversary can lead to stronger lower bounds, and the bound we obtain is therefore tight.

Constructing *difficult* problem instances. Given a problem class $\mathcal{P} = (\mathcal{X}, \mathcal{F}, \phi)$, we construct a "difficult" subclass $\mathcal{F}' = \{f_1, \ldots, f_N\} \subseteq \mathcal{F}$, such that the functions in $\mathcal{F}'$ are hard to distinguish from one another with any possible query sequence, and yet they are sufficiently different from one another so an optimizer for one of them fails to optimize other functions to the same accuracy. With this construction, any algorithm that can reach (ϵ, δ)-accuracy can be used to "differentiate" them if we treat each function as a hypothesis in hypothesis testing. We then consider a fictitious situation in which Nature uniformly selects a function in $\mathcal{F}'$, so that for every algorithm $\mathcal{A}$, we can construct a probability space $(\Omega, \mathcal{B}, \mathbb{P})$ with the following random variables: $M \in \{1, \ldots, N\}$ encodes the random choice of selected function instance in $\mathcal{F}'$; $X^T \in \mathcal{X}^T$ are the queries issued by $\mathcal{A}$ and $\widehat{X}_T \in \mathcal{X}$ is the candidate optimizer[5]; $Y^T \in \mathcal{Y}^T$ are the responses of ϕ to the queries issued by $\mathcal{A}$. The way we construct such $\mathcal{F}'$ is via a "packing set" of the convex domain $\mathcal{X}$.

Suppose given a problem class $\mathcal{P} = (\mathcal{X}, \mathcal{F}, \phi)$, to set up our analysis, given a type of error measure, we first endow the instance space $\mathcal{F}$ with a *distance measure* $\pi(\cdot, \cdot)$ that has the following property: For any $x \in \mathcal{X}$ and any $\epsilon > 0$, and two functions $f, f' \in \mathcal{F}'$, we have

$$\pi(f, f') \geq 2\epsilon \text{ and } \texttt{err}(\widehat{X}_T, f) < \epsilon \Longrightarrow \texttt{err}(\widehat{X}_T, f') \geq \epsilon. \tag{3}$$

In other words, an ϵ-optimizer (whose estimate error with respect to the optima is no larger than ϵ) of a function cannot simultaneously be an ϵ-optimizer of another distinct function. It is easy to construct such distance π satisfying (3) for any particular class $\mathcal{F}$ of continuous functions, and the design of π usually depends on the choice of error measure. For a general $\mathcal{F}$ and function error, we can design π over $\mathcal{F}'$ in the following way: $\pi(f, f') = \inf_{x \in \mathcal{X}} \left[f(x) - \inf_x f(x) + f'(x) - \inf_x f'(x) \right], \forall f, f' \in \mathcal{F}'$. While for point error, we can simply set $\pi(f, f') = \|x_f^* - x_{f'}^*\|$. In the following discussion, we will often implicitly restrict our discussion to a *subclass* of $\mathcal{F}$ and define an appropriate π on that subclass based on the error measure.

Note that at the beginning, Nature will select a function f_M from $\mathcal{F}'$ uniformly at random to be optimized. If one can construct such $\mathcal{F}'$ that satisfies the property specified in Eqn. (3) for a distance measure π, then we are able to show that if any learner's strategy $\mathcal{A}$ achieves a low optimization error over the class $\mathcal{F}'$, then one can use its output to construct an "estimator" $\widehat{M}_T$ that returns the true M of f_M with high probability. So the learner's optimization problem can be reduced to a canonical

[5]We sometimes use $\widehat{X}_T$ instead of $\widehat{X}$ to emphasize its dependency on T.

hypothesis testing problem. We formally prove this after we take into account the requirement of securing the learner's privacy.

Adversary's estimation. We focus on the following class of adversary who will use *proportional-sampling estimators* [23, 19] to infer the optimal point the learner is targeting, where $\widehat{X}^{\mathrm{adv}}$ is sampled from all the queries proportionally. While incorporating a stronger adversary could lead to weaker lower bounds, as we demonstrate later, the lower bound we obtain is actually tight, as it matches the upper bound. In particular, given an observed query sequence X^T, the proportional-sampling estimator is defined as $\widehat{X}^{\mathrm{adv}} = X_t$, where $t \sim \mathsf{Unif}\{1, \ldots, T\}$. Notice that the adversary using proportional-sampling estimator also falls into the class of reasonable adversary. To see this, one can simply treat the adversary's belief as the empirical query distribution, and clearly this belief is consistent. Another way to define proportional-sampling estimator is as follows: The adversary first identifies a $2r$-packing set $\{\theta_1, \ldots, \theta_K\}$ over $\mathcal{X}$ (where $r = \epsilon^{\mathrm{adv}}/L$ for function error and $r = \epsilon^{\mathrm{adv}}$ for point error). For each $k \in [K]$, let $\mathbb{B}(\theta_k, r) = \{x \in \mathcal{X} : \|x - \theta_k\| \leq r\}$ be the ℓ_2-norm ball with the radius of r centering in θ_k. Then depends on the error measure, the proportional-sampling estimator $\widehat{X}^{\mathrm{adv}}$ can also be defined as:

$$\mathbb{P}\left(\widehat{X}^{\mathrm{adv}} = \theta_k\right) = \frac{\sum_{t=1}^{T} \mathbf{1}_{\{X_t \in \mathbb{B}(\theta_k, r)\}}}{T}, \quad k = 1, \ldots, K, \tag{4}$$

where $\mathbf{1}_{\{\mathcal{E}\}}$ is the indicator function of event $\mathcal{E}$. We note that these two methods can coincide with each other when we adopt them to prove the complexity (see the proof of Lemma 1).

Information-theoretical derivations. We now show how to reduce the learner's optimization problem to a canonical hypothesis testing problem, taking into account of securing the learner's privacy. Though our discussions focus on function error, all analysis can be easily adapted to point error. When the context is clear, we suppress r in the notation $\mathbb{B}(\theta_k, r)$ and write it as $\mathbb{B}(\theta_k)$.

Recall that our first step is to construct a subclass of functions $\mathcal{F}' \subseteq \mathcal{F}$ that we use to derive lower bounds. And then, an uniformly selected function $f \in \mathcal{F}'$ is chosen by Nature, and this f will be the learner's unknown objective function. With the adversary's proportional-sampling estimator, the randomness structure leads us to build connections between the adversary's correct estimation probability and the query complexity that we are interested in quantifying. This is summarized in the following lemma.

Lemma 1. *Define the event $\xi_k = \{x_f^* \in \mathbb{B}(\theta_k)\}$. If the adversary follows the proportional-sampling estimator, including the one defined in* (4)*, then to ensure an algorithm is $(\epsilon^{\mathrm{adv}}, \delta^{\mathrm{adv}})$-private, we must have*

$$T_{\mathcal{P}} \geq \frac{1}{\delta^{\mathrm{adv}}} \sum_{t=1}^{T} \mathbb{P}\big(X_t \in \mathbb{B}(\theta_k) \mid \xi_k\big). \tag{5}$$

The above lemma implies that, if we can obtain the lower bound on the right hand side of the above inequality (5), we obtain the lower bound of T, the secure query complexity. In the discussion below, we show that conditional on the event ξ_k, if an algorithm achieves a low minimax error over $\mathcal{F}'$, then one can use its output to construct an estimator $\widehat{M}_T$ that returns the true M most of the time.

Lemma 2. *Suppose an algorithm $\mathcal{A}$ attains a minimax error: $\sup_{f \in \mathcal{F}} \mathbb{P}(\texttt{err}(\widehat{X}_T, f) \geq \epsilon) \leq \delta$. Let $\mathcal{F}' \subseteq \mathcal{F}$ be a finite set $\{f_1, \ldots, f_N\}$ such that every two distinct functions in $\mathcal{F}'$ satisfy* (3)*. Suppose f_M is chosen uniformly at random from $\mathcal{F}'$, and algorithm $\mathcal{A}$ then operates with f_M. Then one can construct $\widehat{M}_T$ for M such that the following holds:*

$$I\left(M; \widehat{M}_T \mid \xi_k\right) \geq (1 - \delta) \log |\mathcal{F}'(\theta_k)| - \log 2 > 0, \tag{6}$$

where $I(\cdot \mid)$ represents the conditional mutual information and $\mathcal{F}'(\theta_k) = \{f_m : x_{f_m}^* \in \mathbb{B}(\theta_k), m \in [N]\}$ denotes the set of functions whose optimizers locate within the ball $\mathbb{B}(\theta_k)$ for a fix $k \in [K]$.

Note that the above mutual information is conditional on the event ξ_k and the inequality holds for every $k \in [K]$. This leads to a critical difference between the above lower bound of mutual information, in which we restrict the number of possible values of $\widehat{M}_T$ to be $|\mathcal{F}'(\theta_k)|$, comparing to that of the non-private one (which should be N). We have thus shown that having a low minimax optimization error over $\mathcal{F}'$ implies that the functions in $\mathcal{F}'$ can be identified most of the time. The above inequality implies that any "good" algorithm of the learner (runs for T steps) should obtain non-trivial amount of information about M at the end of its operation.

On the other hand, the amount of information $I(M; \widehat{M}_T \mid \xi_k)$ is well upper bounded:

Lemma 3. *Fix $k \in [K]$ and for any estimator $\widehat{M}_T : \mathcal{X}^T \times \mathcal{Y}^T \to \{1, \ldots, N\}$, the conditional mutual information can be upper bounded by a summation of two parts:*

$$I(M; \widehat{M}_T \mid \xi_k) \leq \sum_{t=1}^{T} \big(\mathbb{P}\left(X_t \in \mathbb{B}(\theta_k) \mid \xi_k\right) G(X_t \in \mathbb{B}(\theta_k), \xi_k) + \\ \mathbb{P}\left(X_t \notin \mathbb{B}(\theta_k) \mid \xi_k\right) G(X_t \notin \mathbb{B}(\theta_k), \xi_k) \big), \tag{7}$$

where we have $G(X_t \in \mathbb{B}(\theta_k), \xi_k) = \mathbb{E}_M \mathbb{E}_{M'} D_{\mathrm{KL}}\big(\mathbb{P}(Y_t \mid M, X_t \in \mathbb{B}(\theta_k), \xi_k) \| \mathbb{P}(Y_t \mid M', X_t \in \mathbb{B}(\theta_k), \xi_k)\big)$ and $G(X_t \notin \mathbb{B}(\theta_k), \xi_k) = \mathbb{E}_M \mathbb{E}_{M'} D_{\mathrm{KL}}\big(\mathbb{P}(Y_t \mid M, X_t \notin \mathbb{B}(\theta_k), \xi_k) \| \mathbb{P}(Y_t \mid M', X_t \notin \mathbb{B}(\theta_k), \xi_k)\big)$. The expectation $\mathbb{E}_M$ (or $\mathbb{E}_{M'}$) is taken over f_M (or $f_{M'}$) which is uniformly distributed over $\mathcal{F}'(\theta_k)$. And $D_{\mathrm{KL}}\left(\mathbb{P} \| \mathbb{Q}\right)$ denotes the Kullback-Leibler (KL) divergence between $\mathbb{P}$ and $\mathbb{Q}$.

The proof is provided in Appendix A.3. The above lemma characterizes the upper bound of our conditional mutual information via two parts: The first part is the cumulative correct querying probability, while the second one is cumulative incorrect querying probability. Note that in a statistical sense, the divergence $D_{\mathrm{KL}}(\mathbb{P}(Y \mid M, X, \xi_k) \| \mathbb{P}(Y \mid M', X, \xi_k))$ quantifies how close the oracle's responses are for a given query point $x \in \mathcal{X}$ and a given pair $f_M, f_{M'}$ in $\mathcal{F}'(\theta_k)$.

Combining all pieces, we can obtain following general bound which holds for most problem classes.

Theorem 1. *Fix a problem class $\mathcal{P} = (\mathcal{X}, \mathcal{F}, \phi)$ and given an error measure, let $\{\theta_1, \ldots, \theta_K\}$ be a $2r$-packing set over $\mathcal{X}$ (where $r = \epsilon^{\mathrm{adv}}/L$ for function error and $r = \epsilon^{\mathrm{adv}}$ for point error). Suppose there exists a function subclass $\mathcal{F}' \subseteq \mathcal{F}$ such that it satisfies the following conditions:*

1. *the distance measure π defined in Eqn.* (3) *holds for any two distinct functions $f, f' \in \mathcal{F}'$;*
2. *for some $C^* > 0$, $G(x \in \mathbb{B}(\theta_k), \xi_k) \leq C^*, \forall x \in \mathbb{B}(\theta_k)$ and $f_M, f_{M'} \in \mathcal{F}'(\theta_k)$;*
3. *$G(x \notin \mathbb{B}(\theta_k), \xi_k) = 0, \forall x \in \mathbb{B}(\theta_k)$ and $f_M, f_{M'} \in \mathcal{F}'(\theta_k)$.*

Then the secure query complexity satisfies: $T_{\mathcal{P}} \geq \Omega\left(\frac{1-\delta}{C^* \delta^{\mathrm{adv}}} \log |\mathcal{F}'(\theta_k)|\right)$.

Remark 2. The above general lower bound is a direct result of applying Lemma 1 to Lemma 3. Though this lower bound holds generally, it is only tight for certain problem classes. The third condition also provides a hint on how to construct function subclass: Given a coarsening adversary's estimation ball, the functions whose optimizer lie within this ball should be *indistinguishable* based on the function value and gradient information calculated outside this ball.

3.2 Deriving lower bounds

In the section, we demonstrate how to utilize the above analysis for different problem classes. Note that from Theorem 1, the derivation of the lower bounds reduces to finding the problem subclass that satisfies the three listed conditions.

(Noisy) binary search We first explore the secure query complexity of secure binary search $\mathcal{P} = \{[0, 1], \mathcal{F}^{\mathrm{Abs}}, \phi^{\mathsf{sign}}\}$ and secure noisy binary search $\mathcal{P} = \{[0, 1], \mathcal{F}^{\mathrm{Abs}}, \phi^{\mathsf{sign}, p}\}$ as defined in Section 2.1. The result of secure binary search can be summarized as follows.

Theorem 2 (Secure Binary Search). *Given small $\delta, \delta^{\text{adv}} \in (0,1)$, and $2\epsilon \le \epsilon^{\text{adv}} \le \delta^{\text{adv}}/2$,*[6] *for binary search $\mathcal{P} = \{[0,1], \mathcal{F}^{\text{Abs}}, \phi^{\text{sign}}\}$, the secure query complexity is lower bounded as: $T_{\mathcal{P}} \ge \Omega\left(\frac{1-\delta}{\delta^{\text{adv}}} \log(\epsilon^{\text{adv}}/\epsilon)\right)$.*

The full proof of the above theorem is in Appendix B.1.

Remark 3. We obtain the same lower bound as in prior works on secure binary search in the Bayesian setting [23, 19, 21], where a lower bound in the order of $\Omega(\log(\epsilon^{\text{adv}}/\epsilon)/\delta^{\text{adv}})$ was derived. Our use of a different technique based on the minimax analysis allows us to generalize the results to noisy binary search, in which the oracle response is correct with probability p.

Theorem 3 (Secure Noisy Binary Search). *Given small $\delta, \delta^{\text{adv}} \in (0,1)$, and $2\epsilon \le \epsilon^{\text{adv}} \le \delta^{\text{adv}}/2$, for secure noisy binary search $\mathcal{P} = \{[0,1], \mathcal{F}^{\text{Abs}}, \phi^{\text{sign},p}\}$, where $p \in (1/2, 1)$, the secure query complexity is lower bounded as: $T_{\mathcal{P}} \ge \Omega\left(\frac{1-\delta}{\delta^{\text{adv}} c(p)} \log(\epsilon^{\text{adv}}/\epsilon)\right)$, where $c(p) > 0$ is a constant value depending only on the parameter p.*

We defer the detailed proof to Appendix B.2. We obtain a similar bound to the work by [21] for noisy binary search, while their bound contains more refined constants.

Remark 4. For (non-secure) noisy binary search, it is shown [20] that the lower bound of convergence rate is $\mathbb{E}[|x^* - X_T|] = o(c_1^{-T})$ for some constant $c_1 > 1$ depending only on p. Our secure variant converges at the order of $\epsilon^{\text{adv}} c_2^{-T}$, where c_2 is a fixed constant depending on p, δ^{adv}. This is tight up to a multiplicative constant compared with the classic result.

Stochastic convex optimization. We now present our main results for secure stochastic convex optimization. We state our private complexity results with restricting $\mathcal{X}$ to be $[0,1]^d$. Recall that $\mathcal{F}\kappa$ is the set of κ-uniformly convex functions.

Theorem 4 (Secure Stochastic Convex Optimization). *Consider the problem class $\mathcal{P} = [0,1]^d, \mathcal{F}^{\kappa}, \phi^{(1)})$ with a stochastic first-order oracle $\phi^{(1)}$. Then for any $2\sqrt{d}\epsilon \le \epsilon^{\text{adv}} \le (\delta^{\text{adv}})^{1/d}$, small $\delta, \delta^{\text{adv}} \in (0,1)$, the following secure query complexity holds: $T_{\mathcal{P}} \ge \Omega\left(\frac{\sigma^2(\log 2 - h_2(\delta))}{\delta^{\text{adv}} \epsilon^{(2\kappa-2)/\kappa}}\right)$ for function error, $T_{\mathcal{P}} \ge \Omega\left(\frac{\sigma^2(\log 2 - h_2(\delta))}{\delta^{\text{adv}} \epsilon^{2\kappa-2}}\right)$ for point error.*

We defer the proof to Appendix B.3. The key step is to construct a "difficult" function subclass $\mathcal{F}'$ so that the functions in $\mathcal{F}'$ are *indistinguishable* based only the function and gradient information when the query points are outside adversary's estimation region (Condition (3) in Theorem 1).To achieve this, we start with some base convex functions. We then construct the function f in $\mathcal{F}'$ via a maximum operator. This construction helps us ensure the third condition in Theorem 1 is satisfied. An example of the construction when $\kappa = 2$ is given in Fig 1.

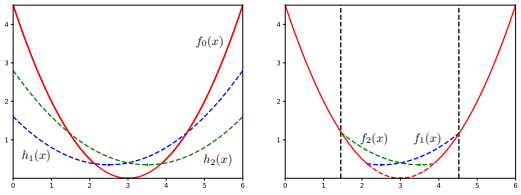


Figure 1: Left: Base functions $f_0(x) = 0.5|x - 3|^2$, $h_1(x) = 0.2|x - (3 - 0.5)|^2 - 1.6$ and $h_2(x) = 0.2|x - (3 + 0.5)|^2 - 1.6$. Right: $f_1(x) = \max\{f_0(x), h_1(x)\}$ and $f_2(x) = \max\{f_0(x), h_2(x)\}$.

We offer a few observations of our results. First, our results match the lower bounds in non-secure convex optimization. In particular, when $\kappa = 2$ (i.e., strongly convex functions), our lower bound matches the known lower bound of standard convex optimization, $\Omega(1/T)$ (because $T_{\mathcal{P}} \ge \Omega(1/\epsilon)$) for function error and $\Omega(1/\sqrt{T})$ for point error. As $\kappa \to \infty$ (i.e., non-strongly convex functions), our lower bound, in the order of $\Omega(1/\sqrt{T})$ for function error, also matches the classic result for Lipschitz convex function optimization. The convergence for point error would fail with non-strongly convex

[6] We restrict the parameter range to exclude trivial cases. For example, if $\epsilon^{\text{adv}} > \delta^{\text{adv}}/2$, the privacy requirement is too strong to be achieved. Consider a naive adversary that obtains an estimate by drawing a point uniformly at random in $[0,1]$. In this case, with probability greater than δ^{adv}, the adversary's estimate is within $\delta^{\text{adv}}/2$ to the optima (due to uniform sampling). If $\epsilon^{\text{adv}} > \delta^{\text{adv}}/2$, the privacy requirement is violated.

functions - this corresponds to the worst case Lipschitz convex functions. As an illustration, it is pointless to "converge" to a single optima for a flat line, a non-strongly convex function.

Second, our privacy constraint leads to a multiplicative penalty of $1/\delta^{\text{adv}}$ in both error measure. This can be considered as a complexity price to pay for the increased privacy. Intuitively, one can also view this penalty as the learner trying to fool the adversary by hiding her non-secure learning strategy within other $\Theta(1/\delta^{\text{adv}})$ fictitiously designed identical strategies.

Third, while our bounds do not seem to explicitly depend on ϵ^{adv}, it is hidden in the logarithmic factor. To be more concrete, according to our Lemma 2, ϵ^{adv} will impact the value of $|\mathcal{F}'(\theta_k)|$, which is bounded by $\Omega\left(\epsilon^{\text{adv}}/\epsilon\right)$. After taking the logarithm to get $\Omega\left(\log(\epsilon^{\text{adv}}/\epsilon)\right)$, we conclude that this term is dominated by $\Omega(1/\epsilon^{\alpha})$ for any $\alpha \geq 1$.

Finally, our results can be extended to the settings with general noisy oracles. As long as Gaussian noise is a subclass of the noise distribution, our lower bounds hold. The Gaussian assumption serves the goal for proving the lower bounds. In the following section, our algorithm and upper bound analysis will also go through for all sub-Gaussian noise oracles. For the ease of presentation, we will focus on Gaussian noise model for the current paper.

4 An Optimal Secure Optimization Strategy

We present a simple and intuitive algorithm that is *optimal* in the sense that it obtains the matching upper bounds in secure query complexity when an arbitrary adversary can present. To secure the learner's privacy, imagine that if the learner performs query uniformly at random for each time step, while the learner sacrifices the learning efficiency, the privacy is secured as no adversary can infer anything from where the leaner queries. The high-level intuition of our algorithm is to mix (secure but non-efficient) uniform query protocol and (efficient but non-secure) standard methods from the optimization literature.

To simplify the presentation, we focus on the one-dimensional case with domain $\mathcal{X} \in [0, 1]$. To be consistent with standard convex optimization algorithms, we present our secure learning protocol where the objective is to optimize the estimation error rate. Inspired by the replicated bisection strategy proposed by Tsitsiklis et al. [19], the general idea of the protocol is as follows: Fixed an oracle budget T, we divide this budget into $\lfloor T/S \rfloor$ phases over each of $S = \left\lfloor 1/\delta^{\text{adv}} \right\rfloor$ many queries. We also divide the domain $[0, 1]$ into equal sub-intervals with length of δ^{adv}. Within each phase, the learner *symmetrically* submits one query to each sub-interval. Among these queries in each phase, there is one query that is consecutively updated according to learner's confidential computation oracle, which can be any efficient algorithm for stochastic convex optimization. The layer of randomization over $\left\lfloor 1/\delta^{\text{adv}} \right\rfloor$-length intervals is the key device to secure the learner's privacy.[7] The details of our secure learning protocol, and together with an example of computation oracle, are included in Appendix B.4.

Below is the formal statement that this secure learning protocol leads to a upper bound of secure query complexity that matches our lower bound up to a logarithmic factor. The proof is in Appendix B.4.

Theorem 5. *Fix $\epsilon^{\text{adv}}, \delta^{\text{adv}} \in (0, 1)$ such that $2\epsilon^{\text{adv}} < \delta^{\text{adv}}$. Learning protocol detailed in Algorithm 1, together with the computation oracle detailed in Algorithm 2, can return an estimator $\widehat{X}_T$ for the learner such that for any $f \in \mathcal{F}^{\kappa}, \kappa > 1$, $f(\widehat{X}_T) - f^* \leq \tilde{\mathcal{O}}\big((T\delta^{\text{adv}})^{-\frac{\kappa}{2\kappa-2}}\big)$ and $|\widehat{X}_T - X^*| \leq \tilde{\mathcal{O}}\big((T\delta^{\text{adv}})^{-\frac{1}{2\kappa-2}}\big)$ hold with probability at least $1 - \delta$. Furthermore, the queries generated from such learning protocol are $(\epsilon^{\text{adv}}, \delta^{\text{adv}})$-private.*

Remark 5. The upper bound holds w.r.t. *arbitrary* adversary strategies. It is easy to verify that the above convergence rate can be translated to an upper bound that matches the lower bound of query complexity. Therefore, our lower bound derived via assuming a specific type of adversary is tight.

[7] Randomization here means that the adversary can't do better by guessing uniformly at random.

5 Discussions and Future Directions

This work studies the secure stochastic convex optimization problem. We present a general information-theoretical analysis and characterize lower bounds. We also give an efficient secure learning protocol with matching upper bounds. A number of open questions remain. In particular, while our current results work for high-dimensional problem instances, we have not analyzed the secure query complexity's dependence on the input dimensions. Characterizing this dependency would be an interesting future direction. In addition, although our lower bound is tight, it relies on assuming the proportional-sampling adversarial strategy. It is unclear whether we can generalize our analysis when considering other certain types of adversaries.

Broader Impact

In this work, we explore the problem of securing the privacy of the learner against a spying adversary. In a broader context, we explore the limit of securing the decision maker's unobservable intent/goal when the query decisions to achieve the intent/goal are observable. Our results, while being theoretical in nature, have potential impacts in providing instructions for designing better security tools to ensure that people's online activities do not create unintended leakage of private information. On the other hand, the discussion on the adversarial strategies could also lead to more delicate attacks, especially to those who are not aware of the existence of attacks from potential adversaries.

Acknowledgments and Disclosure of Funding

We would like to thank Kuang Xu, Jiaming Xu and Dana Yang for the helpful discussions and pointing out the missing assumption of the adversary in Definition 2. We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported in part by ONR Grant N00014-20-1-2240.

References

[1] Alekh Agarwal, Martin J Wainwright, Peter L Bartlett, and Pradeep K Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *Advances in Neural Information Processing Systems*, pages 1–9, 2009.

[2] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. In *Advances in Neural Information Processing Systems*, pages 11282–11291, 2019.

[3] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *Conference On Learning Theory*, volume 3, page 1, 2009.

[4] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):26, 2011.

[5] John Duchi, Feng Ruan, and Chulhee Yun. Minimax bounds on stochastic batched convex optimization. In *Conference On Learning Theory*, pages 3065–3162, 2018.

[6] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

[7] Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 439–449, 2020.

[8] Alon Gonen, Elad Hazan, and Shay Moran. Private learning implies online learning: An efficient reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1): 2489–2512, 2014.

[10] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1, 2012.

[11] Kevin G Jamieson, Robert Nowak, and Ben Recht. Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2012.

[12] Leszek Plaskota. *Noisy information and computational complexity*, volume 95. Cambridge University Press, 1996.

[13] Maxim Raginsky and Alexander Rakhlin. Information-based complexity, feedback and dynamics in convex programming. *IEEE Transactions on Information Theory*, 57(10):7036–7056, 2011.

[14] Aaditya Ramdas and Aarti Singh. Optimal rates for stochastic convex optimization under tsybakov noise condition. In *Proceedings of the 30th International Conference on Machine Learning*, pages 365–373, 2013.

[15] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *Conference On Learning Theory*, 2009.

[16] Alexander Shapiro and Arkadi Nemirovski. On complexity of stochastic programming problems. In *Continuous optimization*, pages 111–146. Springer, 2005.

[17] Wei Tang, Chien-Ju Ho, and Yang Liu. Differentially private contextual dynamic pricing. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1368–1376, 2020.

[18] Abhradeep Guha Thakurta and Adam Smith. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Advances in Neural Information Processing Systems*, pages 2733–2741, 2013.

[19] John Tsitsiklis, Kuang Xu, and Zhi Xu. Private sequential learning. In *Conference On Learning Theory*, 2018.

[20] Rolf Waeber, Peter I Frazier, and Shane G Henderson. Bisection search with noisy responses. *SIAM Journal on Control and Optimization*, 51(3):2261–2279, 2013.

[21] Jiaming Xu, Kuang Xu, and Dana Yang. Optimal query complexity for private sequential learning against eavesdropping. *arXiv preprint arXiv:1909.09836*, 2019.

[22] Jiaming Xu, Kuang Xu, and Dana Yang. Learner-private online convex optimization. In *arxiv,* `https://arxiv.org/abs/2102.11976`, 2021.

[23] Kuang Xu. Query complexity of bayesian private learning. In *Advances in Neural Information Processing Systems*, pages 2431–2440, 2018.

[24] Yuhong Yang and Andrew Barron. Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, pages 1564–1599, 1999.

[25] Bin Yu. Assouad, fano, and le cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.

A Missing Proofs

A.1 Proof of Lemma 1

Proof. For any querying strategy that is $(\epsilon^{\text{adv}}, \delta^{\text{adv}})$-private, it must satisfy $\mathbb{P}(\texttt{err}(\widehat{X}^{\text{adv}}, f) \leq \epsilon^{\text{adv}}) \leq \delta^{\text{adv}}$. We choose function error to prove this lemma. Suppose the adversary's estimator $\widehat{X}^{\text{adv}}$ is obtained through the proportional-sampling, then we have

$$\begin{aligned}\mathbb{P}\left(\texttt{err}(\widehat{X}^{\text{adv}}, f) \leq \epsilon^{\text{adv}}\right) &= \sum_{t=1}^{T} \mathbb{P}\left(\widehat{X}^{\text{adv}} = X_t\right) \mathbb{P}\left(\texttt{err}(X_t, f) \leq \epsilon^{\text{adv}} | \widehat{X}^{\text{adv}} = X_t\right) \\ &\geq \frac{\sum_{t=1}^{T} \mathbb{P}\big(\|X_t - x^*\| \leq \epsilon^{\text{adv}}/L\big)}{T}, \end{aligned} \tag{8}$$

where L is the Lipschitz constant of function f. To ensure $(\epsilon^{\text{adv}}, \delta^{\text{adv}})$-privacy, it deduces that

$$T \geq \frac{1}{\delta^{\text{adv}}} \left(\sum_{t=1}^{T} \mathbb{P}\big(\|X_t - x^*\| \leq \epsilon^{\text{adv}}/L\big) \right). \tag{9}$$

Furthermore, note that f is uniformly-distributed among $\mathcal{F}'$, we have following

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{P}\big(\|X_t - x^*\| \leq \epsilon^{\text{adv}}/L\big) &= \sum_{t=1}^{T} \sum_{k \in [K]} \mathbb{P}\big(\|X_t - x^*\| \leq \epsilon^{\text{adv}}/L \mid \xi_k\big) \cdot \mathbb{P}\big(\xi_k\big) \\ &= \sum_{t=1}^{T} \mathbb{P}\big(X_t \in \mathbb{B}(\theta_k, \epsilon^{\text{adv}}/L) \mid \xi_k\big).\end{aligned}$$

This proves the lemma.

For adversary's strategy defined in (4), the proof is slightly different and we include it below for completeness.

For each $k \in [K]$, let $\Gamma_k = \{X_t : X_t \in \mathbb{B}(\theta_k)\}_{t \geq 1}$ denote the set of queries that lie within the ball $\mathbb{B}(\theta_k, \epsilon^{\text{adv}}/L)$. For the adversary's estimator defined in (4), we also have following reduction to the adversary's probability of correct estimation:

$$\begin{aligned}\mathbb{P}\left(\texttt{err}(\widehat{X}^{\text{adv}}, f) \leq \epsilon^{\text{adv}} | x_f^* \in \mathbb{B}(\theta_k, \epsilon^{\text{adv}}/L)\right) &\geq \mathbb{P}\left(\|\widehat{X}^{\text{adv}} - x_f^*\| \leq \epsilon^{\text{adv}}/L | x_f^* \in \mathbb{B}(\theta_k, \epsilon^{\text{adv}}/L)\right) \\ &= \mathbb{P}\left(\widehat{X}^{\text{adv}} = \theta_k | x_f^* \in \mathbb{B}(\theta_k, \epsilon^{\text{adv}}/L)\right) \\ &= \mathbb{E}\left(\frac{|\Gamma_k|}{\sum_k |\Gamma_k|} \mid x_f^* \in \mathbb{B}(\theta_k, \epsilon^{\text{adv}}/L)\right) \\ &= \frac{\mathbb{E}\left(|\Gamma_k| \mid x_f^* \in \mathbb{B}(\theta_k, \epsilon^{\text{adv}}/L)\right)}{T}.\end{aligned}$$

Note that $|\Gamma_k| = \sum_{t=1} \mathbf{1}_{\{X_t \in \mathbb{B}(\theta_k)\}}$. Thus, we have that

$$\mathbb{P}\left(\texttt{err}(\widehat{X}^{\text{adv}}, f) \leq \epsilon^{\text{adv}} | x_f^* \in \mathbb{B}(\theta_k, \epsilon^{\text{adv}}/L)\right) \geq \frac{\sum_{t=1} \mathbb{P}\big(X_t \in \mathbb{B}(\theta_k, \epsilon^{\text{adv}}/L) \mid \xi_k\big)}{T}.$$

This is the desired result in the lemma.

For point error, the above analysis can be easily carried over by adjusting the term ϵ^{adv}/L to ϵ^{adv}. □

A.2 Proof of Lemma 2

Proof. We note that conditional on event ξ_k, such estimator $\widehat{M}_T$ can be defined as

$$\widehat{M}_T(X^T, Y^T) = \underset{m: f_m \in \mathcal{F}'(\theta_k)}{\operatorname{argmin}} \texttt{err}(\widehat{X}_T, f_m), \tag{10}$$

which simply predicts the function in $\mathcal{F}'$ for which the error of $\widehat{X}_T$ is the smallest. Since $\widehat{X}_T$ is $\sigma(X^T, Y^T)$-measurable, the estimator $\widehat{M}_T$ is indeed a function only of the information available to $\mathcal{A}$ after time T. We define, for each $f_i \in \mathcal{F}'(\theta_k)$, the event

$$\mathcal{E}_i \triangleq \left\{ \texttt{err}(\widehat{X}_T, f_i) \geq \epsilon \right\}.$$

Indeed, if $\mathcal{E}_i$ does not occur, then from the fact that $\pi(f_i, f_j) \geq 2\epsilon$ for all $j \neq i$ and from (3) we deduce that

$$\texttt{err}(\widehat{X}_T, f_j) > \epsilon > \texttt{err}(\widehat{X}_T, f_i), \qquad \forall j \neq i.$$

So it must be the case that $\widehat{M}_T = i$. Therefore,

$$\begin{aligned}\delta &\geq \max_{f_i \in \mathcal{F}'(\theta_k)} \mathbb{P}(\mathcal{E}_i | M = i, \xi_k) \\ &\geq \max_{f_i \in \mathcal{F}'(\theta_k)} \mathbb{P}(\widehat{M}_T \neq i | M = i, \xi_k) \geq \mathbb{P}(\widehat{M}_T \neq M | \xi_k).\end{aligned}$$

In addition, we note that

$$\begin{aligned}\mathbb{P}\left(\texttt{err}(\widehat{X}_T, f) \geq \epsilon\right) &= \sum_k \mathbb{P}\left(\texttt{err}(\widehat{X}_T, f) \geq \epsilon \mid \xi_k\right) \cdot \mathbb{P}(\xi_k) \\ &= \mathbb{P}\left(\texttt{err}(\widehat{X}_T, f) \geq \epsilon \mid \xi_k\right).\end{aligned}$$

Thus, we have $\mathbb{P}(\widehat{M}_T \neq M \mid \xi_k) \leq \delta$. Then by Fano's inequality,

$$\delta \geq \mathbb{P}\left(\widehat{M}_T \neq M \mid \xi_k\right) \geq 1 - \frac{I\left(M; \widehat{M}_T \mid \xi_k\right) + \log 2}{\log |\mathcal{F}'(\theta_k)|}.$$

Rearranging the above inequality will yield us desired result. □

A.3 Proof of Lemma 3

Proof. Our proof is similar to the information radius bound established in [13] where the crux difference is that we mainly operate with the information that is additionally conditional on the event ξ_k. First, note that by chain rule of conditional mutual information, we have

$$\begin{aligned}I\left(M; \widehat{M} \mid \xi_k\right) &\leq I\left(M; X^T, Y^T \mid \xi_k\right) &(11)\\ &= \sum_{t=1}^{T} I\left(M; X_t, Y_t \mid X^{t-1}, Y^{t-1}, \xi_k\right) &(12)\\ &= \sum_{t=1}^{T} I\left(M; X_t \mid X^{t-1}, Y^{t-1}, \xi_k\right) + \sum_{t=1}^{T} I\left(M; Y_t \mid X^{t}, Y^{t-1}, \xi_k\right) &(13)\\ &= \sum_{t=1}^{T} I\left(M; Y_t \mid X^{t}, Y^{t-1}, \xi_k\right), &(14)\end{aligned}$$

where (11) is due to the data processing inequality, (12) and (13) are the chain rule of conditional mutual information, and the last equality (14) is the reason that the choice of X_t is independent of M given the information (X^{t-1}, Y^{t-1}).

Note that for a random triple $(X_1, X_2, X_3) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$, if X_2 and X_3 are conditionally independent given X_1 given $\mathbb{P}$, then the conditional mutual information between X_2 and X_3 given X_1 is defined as:

$$\begin{aligned}I(X_2; X_3 \mid X_1) &= D_{\mathrm{KL}}\left(\mathbb{P}(X_2, X_3 \mid X_1) \| \mathbb{P}(X_2 \mid X_1) \times \mathbb{P}(X_3 \mid X_1) \mid \mathbb{P}(X_1)\right) &(15)\\ &= D_{\mathrm{KL}}\left(\mathbb{P}(X_3 \mid X_1, X_2) \| \mathbb{P}(X_3 \mid X_1) \mid \mathbb{P}(X_1, X_2)\right) &(16)\end{aligned}$$

where (16) is due to the Bayes' rule. Observe that Y_t and M are conditionally independent given the information (X^t, Y^{t-1}, ξ_k), in other words, $M \to (X^t, Y^{t-1}, \xi_k) \to Y_t$ is a Markov chain. Thus, fix some t and consider the conditional mutual information we obtain in (14),

$$\begin{aligned} &I\left(M; Y_t \mid X^t, Y^{t-1}, \xi_k\right) \\ =&D_{\mathrm{KL}}\left(\mathbb{P}(Y_t \mid M, X^t, Y^{t-1}, \xi_k) \middle\| \mathbb{P}(Y^t \mid X^t, Y^{t-1}, \xi_k) \mid \mathbb{P}(M, X^t, Y^{t-1}, \xi_k)\right), \end{aligned} \tag{17}$$

For any estimator $\widehat{M} : X^T \times Y^T \to \{1, \ldots, N\}$, and any sequence of conditional probability measures $\{\mathbb{Q}(Y_t|X^t, Y^{t-1})\}_{t=1}^T$ on $\{\Omega, \mathcal{B}\}$ that satisfying following conditions:

$$\mathbb{P}\left(Y_t \mid X^t, Y^{t-1}\right) \ll \mathbb{Q}\left(Y_t \mid X^t, Y^{t-1}\right), \forall t \in [T], \tag{18}$$

where $\mathbb{P} \ll \mathbb{Q}$ implies that $\mathbb{P}$ is absolute continuous w.r.t. $\mathbb{Q}$. Note that by definition of conditional mutual information, we can write the (17) as follows:

$$\begin{aligned} (17) &= \mathbb{E}\left[\log \frac{d\mathbb{P}(Y_t \mid M, X^t, Y^{t-1}, \xi_k)}{d\mathbb{P}(Y_t \mid X^t, Y^{t-1}, \xi_k)}\right] \\ &= \mathbb{E}\left[\log \frac{d\mathbb{P}(Y_t \mid M, X^t, Y^{t-1}, \xi_k)}{d\mathbb{Q}(Y_t \mid X^t, Y^{t-1}, \xi_k)}\right] - \mathbb{E}\left[\log \frac{d\mathbb{P}(Y_t \mid X^t, Y^{t-1}, \xi_k)}{d\mathbb{Q}(Y_t \mid X^t, Y^{t-1}, \xi_k)}\right] \qquad (19) \\ &= D_{\mathrm{KL}}\left(\mathbb{P}(Y_t \mid M, X^t, Y^{t-1}, \xi_k) \middle\| \mathbb{Q}(Y^t \mid X^t, Y^{t-1}, \xi_k) \mid \mathbb{P}(M, X^t, Y^{t-1}, \xi_k)\right) - \\ &\quad D_{\mathrm{KL}}\left(\mathbb{P}(Y_t \mid X^t, Y^{t-1}, \xi_k) \middle\| \mathbb{Q}(Y^t \mid X^t, Y^{t-1}, \xi_k) \mid \mathbb{P}(X^t, Y^{t-1}, \xi_k)\right) \qquad (20) \\ &\leq D_{\mathrm{KL}}\left(\mathbb{P}(Y_t \mid M, X^t, Y^{t-1}, \xi_k) \middle\| \mathbb{Q}(Y^t \mid X^t, Y^{t-1}, \xi_k) \mid \mathbb{P}(M, X^t, Y^{t-1}, \xi_k)\right) \qquad (21) \end{aligned}$$

where (19) and (20) are from the condition (18), and (21) is due to the fact that the mutual information are non-negative. Taking the summation over time t, we obtain that:

$$\begin{aligned} I(M; \widehat{M} \mid \xi_k) &\leq \sum_{t=1}^{T} D_{\mathrm{KL}}\big(\mathbb{P}\big(Y_t \mid M, X^t, Y^{t-1}, \xi_k\big) \| \mathbb{Q}\big(Y_t \mid X^t, Y^{t-1}, \xi_k\big) \mid \mathbb{P}\big(M, X^t, Y^{t-1}, \xi_k\big)\big) \\ &= \sum_{t=1}^{T} D_{\mathrm{KL}}\left(\mathbb{P}(Y_t \mid M, X_t, \xi_k) \middle\| \mathbb{Q}\left(Y_t \mid M', X_t, \xi_k\right) \mid \mathbb{P}\left(M, X^t, Y^{t-1}, \xi_k\right)\right), \end{aligned} \tag{22}$$

where (22) is by hypothesis on oracle's behavior: $(X^{t-1}, Y^{t-1}) \to (M, X_t) \to Y_t$ is a Markov chain. Thus, we can write $\mathbb{P}(Y_t \mid M, X^t, Y^{t-1}, \xi_k)$ as $\mathbb{P}(Y_t \mid M, X_t, \xi_k)$.

At each round t, take $\mathbb{Q}\left(Y_t \mid X^t, Y^{t-1}, \xi_k\right) = \mathbb{Q}\left(Y_t \mid X_t, \xi_k\right)$, and if we set $\mathbb{Q}\left(Y_t \mid X_t, \xi_k\right)$ to be $\mathbb{P}(Y_t|M, X_t, \xi_k)$ with f_M uniformly distributed in $\mathcal{F}'(\theta_k)$, we will have following:

$$\begin{aligned} \mathbb{Q}\left(Y_t \mid X_t, \xi_k\right) &= \frac{1}{|\mathcal{F}'(\theta_k)|} \sum_{i \in \mathcal{F}'(\theta_k)} \mathbb{P}\left(Y_t \mid M = i, X_t, \xi_k\right) \\ &= \mathbb{E}_M \mathbb{P}\left(Y_t \mid M, X_t, \xi_k\right). \end{aligned}$$

Then, introducing an independent copy of M (M'), and noting that $\mathbb{Q}\left(Y_t \mid X_t, \xi_k\right) = \mathbb{E}_{M'} \mathbb{P}\left(Y_t \mid M', X_t, \xi_k\right)$, we can obtain following upper bound of the conditional mutual information

we are operating on:

$$I(M;\widehat{M}\mid\xi_k)\le\sum_{t=1}^{T}\mathbb{E}_{M'}D_{\mathrm{KL}}\left(\mathbb{P}(Y_t\mid M,X_t,\xi_k)\|\mathbb{P}(Y_t\mid M',X_t,\xi_k)\mid\mathbb{P}(M,X_t,\xi_k)\right) \tag{23}$$

$$=\sum_{t=1}^{T}\mathbb{E}_{M,X_t,\xi_k}\mathbb{E}_{M'}D_{\mathrm{KL}}\left(\mathbb{P}(Y_t\mid M,X_t,\xi_k)\|\mathbb{P}(Y_t\mid M',X_t,\xi_k)\right)$$

$$=\sum_{t=1}^{T}\sum_{M,X_t,\xi_k}\mathbb{P}\left(M\mid X_t,\xi_k\right)\mathbb{P}\left(X_t\mid\xi_k\right)\mathbb{P}\left(\xi_k\right)\mathbb{E}_{M'}D_{\mathrm{KL}}\left(\mathbb{P}(Y_t\mid M,X_t,\xi_k)\|\mathbb{P}(Y_t\mid M',X_t,\xi_k)\right)$$

$$=\sum_{t=1}^{T}\sum_{x\in\mathcal{X}}\mathbb{P}\left(X_t=x\mid\xi_k\right)\sum_{M,\xi_k}\mathbb{P}\left(M\mid X_t=x,\xi_k\right)\mathbb{P}\left(\xi_k\right)\cdot$$
$$\mathbb{E}_{M'}D_{\mathrm{KL}}\big(\mathbb{P}(Y_t\mid M,X_t=x,\xi_k)\|\mathbb{P}(Y_t\mid M',X_t=x,\xi_k)\big)$$

$$=\sum_{t=1}^{T}\Big(\mathbb{P}\left(X_t\in\mathbb{B}(\theta_k)\mid\xi_k\right)\mathbb{E}_M\mathbb{E}_{M'}D_{\mathrm{KL}}\big(\mathbb{P}(Y_t\mid M,X_t\in\mathbb{B}(\theta_k),\xi_k)\|\mathbb{P}(Y_t\mid M',X_t\in\mathbb{B}(\theta_k),\xi_k)\big)+$$
$$\mathbb{P}\left(X_t\notin\mathbb{B}(\theta_k)\mid\xi_k\right)\mathbb{E}_M\mathbb{E}_{M'}D_{\mathrm{KL}}\big(\mathbb{P}(Y_t\mid M,X_t\notin\mathbb{B}(\theta_k),\xi_k)\|\mathbb{P}(Y_t\mid M',X_t\notin\mathbb{B}(\theta_k),\xi_k)\big)\Big), \tag{24}$$

where we have used the convexity property of the divergence and inequality (23) is then the result of Jensen's inequality. The last equality (24) used the fact that Nature selects function f uniformly at random. The expectation $\mathbb{E}_M$ (or $\mathbb{E}_{M'}$) is taken over f_M (or $f_{M'}$) which is uniformly distributed over $\mathcal{F}'(\theta_k)$. □

B Proofs for main results

B.1 Proofs for Theorem 2

Proof. Let $\Lambda_\epsilon=\{\alpha_1,\ldots,\alpha_N\}$ and $\Lambda_{\epsilon^{\mathrm{adv}}}=\{\theta_1,\ldots,\theta_K\}$ denote maximal 2ϵ-packing set, $2\epsilon^{\mathrm{adv}}$-packing set in $[0,1]$, respectively. We define following function subclass $\mathcal{F}'=\{f_\alpha(x)\}_{\alpha\in\Lambda_\epsilon}\subset\mathcal{F}^{\mathrm{Abs}}$:

$$f_\alpha(x)=|x-\alpha|,\quad\alpha\in\Lambda_\epsilon. \tag{25}$$

It is easy to see that, $N\ge1/\epsilon$ and $K\ge1/\epsilon^{\mathrm{adv}}$. Furthermore, we also have $\pi(f_{\alpha_i},f_{\alpha_j})=|\alpha_i-\alpha_j|\ge2\epsilon$. Now let f_{α_M} be the function selected by Nature among $\mathcal{F}'$, and recall that ξ_k denotes the event $\{\alpha_M\in[\theta_k-\epsilon^{\mathrm{adv}},\theta_k+\epsilon^{\mathrm{adv}}]\}$. Then, by Lemma 2, we have

$$I(M;\widehat{M}_T|\xi_k)\ge(1-\delta)\log|\mathcal{F}'(\theta_k)|-\log2, \tag{26}$$

where $|\mathcal{F}'(\theta_k)|=\epsilon^{\mathrm{adv}}/\epsilon$ by construction. On the other hand, we can also upper bound the above conditional mutual information. From the fact $I(X;Y|Z)=H(X|Z)-H(X|Y,Z)$ and entropy $H(\cdot)$ is nonnegative, we have $I(X;Y|Z)\le H(X|Z)$, thus

$$I(M;\widehat{M}_T|\xi_k)\le H(Y^T|\xi_k) \tag{27}$$
$$\le H(Y_1|\xi_k)+\sum_{t=1}^{T-1}H(Y_{t+1}|Y_1,\ldots,Y_t,\xi_k)\qquad\text{(By chain rule)}$$

Note that, by definition, we have

$$H(Y_{t+1}|Y_1,\ldots,Y_t,\xi_k)=\sum_{y_1,\ldots,y_t}\mathbb{P}(Y_1=y_1,\ldots,Y_t=y_t|\xi_k)H(Y_{t+1}|Y_1=y_1,\ldots,Y_t=y_t,\xi_k) \tag{28}$$

Observe that, conditional on the event ξ_k, if an algorithm $\mathcal{A}_t(y_1, \ldots, y_t)$ outputs the next query X_{t+1} which is smaller than $\theta_k - \epsilon^{\text{adv}}$, then we must have $Y_{t+1} = -1$, while if it is larger than $\theta_k + \epsilon^{\text{adv}}$, then we have $Y_{t+1} = +1$. Moreover when X_{t+1} is in the range $[\theta_k - \epsilon^{\text{adv}}, \theta_k + \epsilon^{\text{adv}}]$, $H(Y_{t+1}|\cdot)$ can take only two values, namely $+1$ and -1. Thus, $H(Y_{t+1}|X_{t+1} \in [\theta_k - \epsilon^{\text{adv}}, \theta_k + \epsilon^{\text{adv}}]) \leq 1$. The above observations give us following result

$$\sum_{y_1,\ldots,y_t} \mathbb{P}(Y_1 = y_1, \ldots, Y_t = y_t|\xi_k) H(Y_{t+1}|Y_1 = y_1, \ldots, Y_t = y_t, \xi_k) \tag{29}$$

$$= \sum_{y_1,\ldots,y_t : \mathcal{A}_t(y_1,\ldots,y_t) \in \mathbb{B}(\theta_k)} \mathbb{P}(Y_1 = y_1, \ldots, Y_t = y_t|\xi_k) H(Y_{t+1}|Y_1 = y_1, \ldots, Y_t = y_t, \xi_k) \tag{30}$$

$$\leq \mathbb{P}(X_{t+1} \in \mathbb{B}(\theta_k)|\xi_k). \tag{31}$$

With inequality in (26), we conclude our result. □

B.2 Proofs for Theorem 3

Proof. The proof is overall similar to the one in secure binary search, we also construct two packing sets Λ_ϵ and $\Lambda_{\epsilon^{\text{adv}}}$ to set up our analysis. The only difference is how we bound the KL divergence of two probability measures induced by two randomly selected function instances in $\mathcal{F}'$. In particular, note that conditional on event ξ_k, i.e., $x^* \in \mathbb{B}(\theta_k) = [\theta_k - \epsilon^{\text{adv}}, \theta_k + \epsilon^{\text{adv}}]$, for any function f in $\mathcal{F}'(\theta_k)$, when the query X_t is smaller than $\theta_k - \epsilon^{\text{adv}}$, we have $\mathbb{P}(Y_{t+1} = -1) = p$ and $\mathbb{P}(Y_{t+1} = +1) = 1 - p$; while the query X_t is larger than $\theta_k + \epsilon^{\text{adv}}$, we have $\mathbb{P}(Y_{t+1} = +1) = p$ and $\mathbb{P}(Y_{t+1} = -1) = 1 - p$. One of important observations is when the query is outside of $\mathbb{B}(\theta_k)$, the gradient information provided by the oracle will have the same probability measure for all functions in $\mathcal{F}'$. This implies

$$D_{\text{KL}}\big(\mathbb{P}(Y_t \mid M, X_t \notin \mathbb{B}(\theta_k), \xi_k) \| \mathbb{P}(Y_t \mid M', X_t \notin \mathbb{B}(\theta_k), \xi_k)\big) = 0, \quad \forall f_M, f_{M'} \in \mathcal{F}'(\theta_k). \tag{32}$$

On the other hand, when $X_t \in \mathbb{B}(\theta_k)$, for any $f_M, f_{M'} \in \mathcal{F}'(\theta_k)$, we can upper bound the KL divergence as follows:

$$D_{\text{KL}}\big(\mathbb{P}(Y_t \mid M, X_t \in \mathbb{B}(\theta_k), \xi_k) \| \mathbb{P}(Y_t \mid M', X_t \in \mathbb{B}(\theta_k), \xi_k)\big) = p \log \frac{p}{1-p} + (1-p) \log \frac{1-p}{p}.$$

Thus, according to Lemma 3, we have:

$$I(M; \widehat{M}_T|\xi_k) \leq c(p) \sum \mathbb{P}(X_t \in \mathbb{B}(\theta_k)|\xi_k), \tag{33}$$

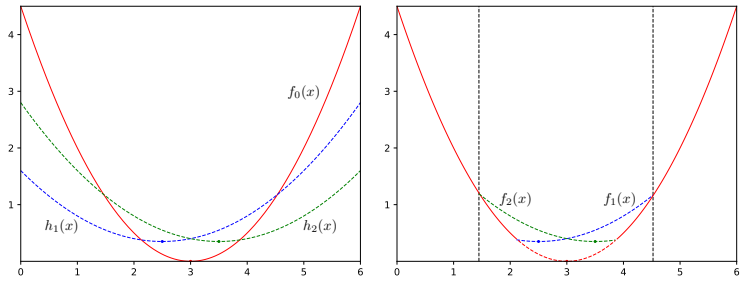
where $c(p) = (2p-1)\log(p/(1-p))$. Putting together the pieces yields our result. □

B.3 Proofs for Theorem 4

Proof. We first prove the result for point error, the result of function error can be achieved by a Jensen's inequality (please see the end of the proof). The general technique of our proof is rather similar to that of statistical minimax analysis for oracle complexity in stochastic convex optimization, but the construction here is a bit more intricate. Specifically, we will pick two similar functions $\mathcal{F}' = \{f_1, f_2\}$ in the class $\mathcal{F}$ and show that they are hard to differentiate with only T queries to the oracle $\phi^{(1)}$. A significant difference to the standard minimax function construction, as will be shown shortly, is that the way how we construct such f_1 and f_2: our goal is to make the information gain on differentiating f_1 and f_2 will be zero as long as the learner queries the points a bit far from optimal points. In particular, consider the domain $\mathcal{X} = [0,1]^d$, we first define following base functions, which will be used for us to construct f_1 and f_2: $f_0(x) = c_0\|x - x_0^*\|^\kappa$; $h_1(x) = c_1\|x - (x_0^* - \epsilon/\sqrt{d} \cdot \mathcal{I}_d)\|^\kappa + c_2$, and $h_2(x) = c_1\|x - (x_0^* + \epsilon/\sqrt{d} \cdot \mathcal{I}_d)\|^\kappa + c_2$, where $x_0^* = (1/2, \ldots, 1/2)$. We now define functions f_1 and f_2 as follows:

$$f_1(x) = \max\{f_0(x), h_1(x)\}; \quad f_2(x) = \max\{f_0(x), h_2(x)\}, \tag{34}$$

where c_0, c_1 are constants ensuring f_1 and f_2 are L-Lipschitz. Convexity is maintained by the maximum operator over two convex functions. Let x' be one of the solutions for $f_0(x) = h_2(x)$,



(a) (b)

Figure 2: A illustration for construction of Convex functions when $\kappa = 2$ and $d = 1$. (a) We use functions $f_0(x) = 0.5|x - 3|^2$, $h_1(x) = 0.2|x - (3 - 0.5)|^2 - 1.6$ and $h_1(x) = 0.2|x - (3 + 0.5)|^2 - 1.6$ as base functions. (b) We then construct $f_1(x) = \max\{f_0(x), h_1(x)\}$ and $f_2(x) = \max\{f_0(x), h_2(x)\}$. In this plot, we choose these numerical constants to ensure that the functions f_1 and f_2 are indistinguishable based only the function and gradient information when the query points are outside adversary's estimation region.

which x' should depend on the constant c_2. We now chose c_2 to satisfy following condition: $\|x_0^* - x'\| \geq \epsilon^{\text{adv}}$. By and large, f_1 are f_2 are constructed such that the learner has to *strenuously* nail down her search within a region which is near to the minimizer. Note that, even though we only construct two functions in $\mathcal{F}'$, we can still ensure that each estimation ball $\mathbb{B}(\theta_k)$ (e.g., when $d = 1$, the subinterval $[2(k-1)\epsilon^{\text{adv}}, 2k\epsilon^{\text{adv}}]$), for adversary, contain the same number of hypothesis functions we construct. To see this, we can just add one more randomization before the Nature draws function $f \in \mathcal{F}$. In particular, we can just replicate a same function subclass for each estimation ball by translating the above $\mathcal{F}'$ along the domain $\mathcal{X}$. Thus, the Nature can just first uniformly sample a function subclass, then sample a function f from that subclass. By construction, we can ensure the quantity $\mathcal{F}'(\theta_k) = 2$ for each estimation ball $\mathbb{B}(\theta_k)$.

Also, note that by triangle inequality, upon defining $\pi(f_1, f_2) = \|x_{f_1}^* - x_{f_2}^*\|$ will guarantee us the property in (3). Moreover, let J denote the region $J = \{x : x \in \mathbb{B}(x_0^*, \|x_0^* - x'\|)\}$ which may contain the ball $\mathbb{B}(\theta_k)$ (this is by our condition for c_2). Noticeably, the function $f_1(x)$ and $f_2(x)$ are different only within the region J, while they are indistinguishable based only on function value and gradient information calculated outside J.

We now proceed to utilize the information bounds we derive in earlier sections to prove our main result. Note that, by construction, at most two functions whose x^*s will locate in the region J, same for $\mathbb{B}(\theta_k)$. Thus, given the realized selected function index $M \in \{1, 2\}$, by Fano's inequality, we have

$$I(M; \widehat{M}_T | \xi_k) \geq \log 2 - h_2(\delta), \tag{35}$$

where $h_2(\delta) := -\delta \log \delta - (1 - \delta) \log(1 - \delta)$ is the binary entropy function. Let $\mathbb{Q} = \frac{1}{2} \sum_{i=1}^{2} \mathbb{P}(Y_t | M = i, X_t)$, we then have

$$I(M; \widehat{M}_T | \xi_1) \leq \sum_{t=1}^{T} \sum_{x \in \mathcal{X}} \mathbb{P}(X_t = x) \cdot \mathbb{E}_{M,M'} D_{\text{KL}} \left(\mathbb{P}(Y_t | M, X_t = x, \xi_k) \| \mathbb{P}(Y_t | M', X_t = x, \xi_k) \right).$$

Note that by Lemma 3, the RHS of the above inequality can be divided into two parts: one is for summation over $x \notin \mathbb{B}(\theta_k)$, while another is for $x \in \mathbb{B}(\theta_k)$.

By construction we know that f_1 and f_2 are indistinguishable when $x \notin J$, the same holds for $x \in \mathbb{B}(\theta_k)$. Thus, the learner will obtain no information on which function she is optimizing if her queries are outside of the domain J. In other words, the KL divergence will equal to zero when $x \notin J$:

$$D_{\text{KL}} \left(\mathbb{P}(Y_t | M, X_t = x, \xi_k) \| \mathbb{P}(Y_t | M', X_t = x, \xi_k) \right) = 0, \quad \forall x \notin J. \tag{36}$$

We now proceed to bound the divergence $D_{\mathrm{KL}}(\mathbb{P}(Y \mid M, X, \xi_k) \| \mathbb{P}(Y \mid M', X, \xi_k))$ when $x \in J$. Recall that the response from the oracle at the query point x contains the value of $f(x)$ and its gradient information at x: $g(x)$. In particular, let $y_1 = f(x) + w_1$ and $y_2 = g(x) + w_2$ denote the noisy function value and noisy gradient value, respectively. Then y_1 and y_2 are conditionally independent given $M = i$ and $X = x$, for the Gaussian oracle, they can be represented as follows:

$$\mathbb{P}(Y \mid M = i, X = x, \xi_k) = \mathbb{P}(y_1 \mid M = i, X = x, \xi_k) \cdot \mathbb{P}(y_2 \mid M = i, X = x, \xi_k),$$

where $\mathbb{P}(y_1 \mid M = i, X = x, \xi_k) = \mathsf{N}(f_i(x), \sigma^2)$ and $\mathbb{P}(y_2 \mid M = i, X = x, \xi_k) = \mathsf{N}(g_i(x), \sigma^2 \mathcal{I}_d)$, and $\mathcal{I}_d$ denotes a $d-$dimensional identity vector. Therefore, we can bound the divergence

$$\begin{aligned}
&D_{\mathrm{KL}}\big(\mathbb{P}(Y|M = i, X = x, \xi_k) \| \mathbb{P}(Y|M = j, X = x, \xi_k)\big) \\
=&D_{\mathrm{KL}}\big(\mathbb{P}(y_1|M = i, X = x, \xi_k) \| \mathbb{P}(y_1|M = j, X = x, \xi_k)\big) + \\
&D_{\mathrm{KL}}\big(\mathbb{P}(y_2|M = i, X = x, \xi_k) \| \mathbb{P}(y_2|M = j, X = x, \xi_k)\big) \\
=&D_{\mathrm{KL}}\left(\mathsf{N}(f_i(x), \sigma^2) \big\| \mathsf{N}(f_j(x), \sigma^2)\right) + D_{\mathrm{KL}}\left(\mathsf{N}(g_i(x), \sigma^2 \mathcal{I}_d) \big\| \mathsf{N}(g_j(x), \sigma^2 \mathcal{I}_d)\right) \\
=&\frac{1}{2\sigma^2}\left([f_i(x) - f_j(x)]^2 + \|g_i(x) - g_j(x)\|^2\right).
\end{aligned}$$

Take the supreme over $\mathcal{X}$ and all possible (i, j) conditional on the event ξ_k will yield us following:

$$\begin{aligned}
&D_{\mathrm{KL}}\big(\mathbb{P}(Y|M = i, X = x, \xi_k) \| \mathbb{P}(Y|M = j, X = x, \xi_k)\big) \\
\leq &\sup_{\substack{x \in \mathcal{X}; \\ f_i, f_j \in \mathcal{F}'(\theta_k)}} \frac{1}{2\sigma^2}\left([f_i(x) - f_j(x)]^2 + \|g_i(x) - g_j(x)\|^2\right).
\end{aligned}$$

Thus, back to Lemma 3, we have

$$\begin{aligned}
&I(M; \widehat{M}_T|\xi_k) \\
\leq &\sum_{t=1}^{T} \mathbb{P}(\|X_t - x^*\| \leq \epsilon^{\mathrm{adv}}) \cdot \max_{x \in J} \frac{[f_1(x) - f_2(x)]^2 + \|g_1(x) - g_2(x)\|^2}{\sigma^2} \\
\leq &\frac{c_1^2}{\sigma^2} \max_{x \in J} \left(\left(\left\| x_0^* - \frac{\epsilon \mathcal{I}_d}{\sqrt{d}} \right\|^{\kappa} - \left\| x_0^* + \frac{\epsilon \mathcal{I}_d}{\sqrt{d}} \right\|^{\kappa} \right)^2 + \kappa^2 \left(\left\| x_0^* - \frac{\epsilon \mathcal{I}_d}{\sqrt{d}} \right\|^{\kappa - 1} - \left\| x_0^* + \frac{\epsilon \mathcal{I}_d}{\sqrt{d}} \right\|^{\kappa - 1} \right)^2 \right) \cdot \\
&\qquad \sum_{t=1}^{T} \mathbb{P}(\|X_t - x^*\| \leq \epsilon^{\mathrm{adv}}) \\
\leq &\mathcal{O}\left(\frac{\epsilon^{2\kappa - 2}}{\sigma^2}\right) \sum_{t=1}^{T} \mathbb{P}(\|X_t - x^*\| \leq \epsilon^{\mathrm{adv}}).
\end{aligned}$$

As a consequence, we have following:

$$\sum_{t=1}^{T} \mathbb{P}(\|X_t - x^*\| \leq \epsilon^{\mathrm{adv}}) \geq \mathcal{O}\left(\frac{\sigma^2(\log 2 - h_2(\delta))}{\epsilon^{2\kappa - 2}}\right) \tag{37}$$

Putting together our bounds with the Equation (9) will give us desired secure oracle complexity for point error. For the result of function error, note that given $\kappa > 1$ we have

$$\begin{aligned}
\inf_{\mathcal{A}} \sup_{f \in \mathcal{F}'} \mathbb{E}\left[|f(\widehat{X}_T) - f^*|\right] &\geq \inf_{\mathcal{A}} \sup_{f \in \mathcal{F}'} \mathbb{E}[\lambda \|\widehat{X}_T - x_f^*\|^{\kappa}]. &(38)\\
&\geq \inf_{\mathcal{A}} \sup_{f \in \mathcal{F}'} \mathbb{E}[\lambda \|\widehat{X}_T - x_f^*\|]^{\kappa}. &\text{(By Jensen's inequality)}
\end{aligned}$$

Invoking Markov inequality will give us the secure oracle complexity for function error. □

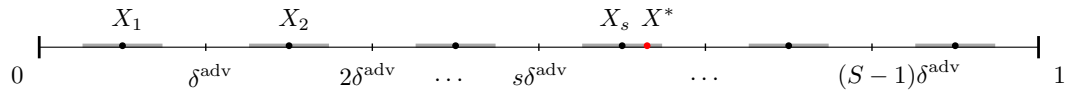


Figure 3: A graph illustration on Algorithm 1. The length of each shade interval is 2ϵ. The black dots $\{X_1, X_2, \ldots\}$ are the queries for last phase, while the red dot is the location of minimizer.

Algorithm 1 Secure Learning Protocol

Input: $S := \lfloor 1/\delta^{\mathrm{adv}} \rfloor$, $K := \lfloor T/S \rfloor$, exponent $\kappa > 0$, convexity parameter $\lambda > 0$, confidence $\delta > 0$, subgradient bound W.
Initialize $x_1 \in [0,1]$ arbitrarily and set $\mathcal{G}_1 = \{g_1\}$, divide $[0,1]$ into subintervals with the equal length of being δ^{adv}.
for $k = 1, \ldots, K$ **do**
  Let $\bar{x} = \mathrm{EpochGD}(\kappa, \lambda, \delta, W, K, \mathcal{G}_k)$.
  for $i \in [S]$ **do**
    $s \sim \mathtt{Uniform}\{1, 2, \ldots, S\}$.
    Query the oracle at the point $x_{(k-1)S+s+1} = (s-1)\delta^{\mathrm{adv}} + (\bar{x} - \big(J(\bar{x}, \delta^{\mathrm{adv}}) - 1\big)\, \delta^{\mathrm{adv}})$.
    if $s = J(\bar{x}, \delta^{\mathrm{adv}})$ **then**
      Record the gradient $g_{(k-1)S+s+1}$ obtained from the oracle: $\mathcal{G}_k \leftarrow \mathcal{G}_k \cup \{g_{(k-1)S+s+1}\}$.
    end if
  end for
end for
Output: Learner's estimation: $\bar{x}$.

B.4 Algorithm and the Proof for Theorem 5

For notational simplicity, let $J(x, \delta^{\mathrm{adv}})$ denote the index of subinterval which contains the point x when $[0,1]$ is uniformly divided in subintervals with the length of $\lfloor 1/\delta^{\mathrm{adv}} \rfloor$ and let $K = \lfloor T/S \rfloor$.

Proof. We now establish the privacy guarantee when the adversary's error measure is point error. The proof can be similarly carried over to function error. Recall that the learner actually performs parallel EpochGD on the S subintervals $\{((s-1)\delta^{\mathrm{adv}}, s\delta^{\mathrm{adv}}]\}_{s\in[S]}$. Since the adversary only observes the queries, and he is not aware of the learner's confidential computation oracle, he learns that X^* is contained in one of these S subintervals. Moreover, due to the strictly symmetrical querying over these subintervals, the adversary also cannot tell which of the subintervals contains X^*. Specifically, let $\{X_s\}_{s\in[S]}$ denote the learner's last phase queries. Then the adversary knows following:

$$\frac{1-\delta}{S} \le \mathbb{P}(|X_s - X^*| \le \epsilon) \le 1, \quad \forall s \in [S].$$

Thus, at the end of the last phase, the adversary will know that X^* belongs to one of the subintervals $\{[X_s - \epsilon, X_s + \epsilon]\}_{s\in[S]}$ with high probability, where $\epsilon = \tilde{\mathcal{O}}\big((T\delta^{\mathrm{adv}})^{-\frac{1}{2\kappa-2}}\big)$. Recall that the adversary is endowed with an uniform prior knowledge on where X^* is, then it can be computed that the adversary's posterior density of X^* is:

$$f_{X^*}(x|\text{queries}) = \begin{cases} (1-\delta)/(2S\epsilon), & \forall x \in \cup_{s=1}^{S}[X_s - \epsilon, X_s + \epsilon] \\ \delta/(1-2S\epsilon), & \text{o.w.} \end{cases} \tag{39}$$

Since δ is a small value (i.e., $\ll 0.5$), thus, for any subinterval $\mathcal{L} \subset [0,1]$ with the length of $2\epsilon^{\mathrm{adv}}$, it is adversary's best strategy to narrow down his estimation region which could cover one of subintervals $\{[X_s - \epsilon, X_s + \epsilon]\}_{s\in[S]}$. Now, let $\mu(\cdot)$ denote the Lebesgue measure of subsets of $[0,1]$, note that

$$\mu\left(\mathcal{L} \cap \cup_{s=1}^{S}[X_s - \epsilon, X_s + \epsilon]\right) \le 2\epsilon.$$

Together with the Eqn. (39), we find that, for any adversary's estimator $\widehat{X}^{\mathrm{adv}}$, we have

$$\mathbb{P}(|\widehat{X}^{\mathrm{adv}} - X^*| \le \epsilon^{\mathrm{adv}}|\text{queries}) \le \frac{1-\delta}{2S\epsilon} \cdot 2\epsilon + \frac{\delta}{1-2S\epsilon} \cdot (2\epsilon^{\mathrm{adv}} - 2\epsilon). \tag{40}$$

Algorithm 2 EpochGD (κ, λ, δ, W, T, $\mathcal{G}$)

Initialize $x_1^1 = x_1$, $e = t = 1$.
Initialize $T_1 = 2C_0$, $\eta_1 = C_1\, 2^{-\frac{\kappa}{2\kappa-2}}$, $R_1 = \left(\frac{C_2\eta_1}{\lambda}\right)^{1/\kappa}$.
while $\sum_{i=1}^{e} T_i \le T$ **do**
  if $|\mathcal{G}| < \sum_{i=1}^{e} T_i$ **then**
    Get the newest element in $\mathcal{G}$, denote it by g_t.
    Set $\mathcal{K} := [0,1] \cap [x_1^e - R_e, x_1^e + R_e]$.
    Output: $x_{t+1}^e = \operatorname{argmin}_{x\in\mathcal{K}} |(x_t^e - \eta_e g_t) - x|$. ▷ Return the value to protocol
    Update: $\mathcal{G}$.
    Set $t \leftarrow t+1$.
  else
    Set $x_1^{e+1} = \frac{1}{T_e}\sum_{t=1}^{T_e} x_t^e$.
    Output: x_1^{e+1}. ▷ Return the value to protocol
    Update: $\mathcal{G}$.
    Set $T_{e+1} = 2T_e$, $\eta_{e+1} = \eta_e \cdot 2^{-\frac{\kappa}{2\kappa-2}}$.
    Set $R_{e+1} = \left(\frac{C_2\eta_{e+1}}{\lambda}\right)^{1/\kappa}$, $e \leftarrow e+1$, $t = 1$.
  end if
end while
Output: x_1^e.

Under the assumption that $2\epsilon^{\mathrm{adv}} < \delta^{\mathrm{adv}}$, the RHS of Eqn. (40) will be smaller than $1/S$. We thus establish the privacy guarantee for any adversary's estimators.

We prove the accuracy guarantee of the above secure learning protocol with appropriate chosen C_0, C_1, C_2. Specifically, set $C_0 = 288\log(\lfloor \log T\delta^{\mathrm{adv}} + 1 \rfloor / \delta)$, $C_1 = \frac{G^{\frac{2-\kappa}{\kappa-1}} 2^{\frac{\kappa}{2(\kappa-1)^2}}}{\lambda^{1/(\kappa-1)}}$, $C_2 = 2^{\frac{\kappa}{2\kappa-2}} W^2$. Follow the analysis of [9, 14], we know that given a total oracle budge T with dividing into a series of consecutive epochs $\{T_1, 2T_1, \ldots, 2^e T_1, \ldots\}$, and running standard stochastic gradient descent in each epoch, will ensure us $f(\widehat{X}_T) - f^* \le \tilde{\mathcal{O}}\big(T^{-\frac{\kappa}{2\kappa-2}}\big)$ and $|\widehat{X}_T - X^*| \le \tilde{\mathcal{O}}\big(T^{-\frac{1}{2\kappa-2}}\big)$ hold with probability at least $1-\delta$ for some estimator $\widehat{X}_T$. Thus, adapted to our setting, our total oracle budget is $\lfloor T\delta^{\mathrm{adv}} \rfloor$. Plugging this into the above results will help us to get the accuracy guarantee. As a sanity check, one can also verify that the error rate presented in our Theorem 5 can be easily translated to match our oracle complexity in Theorem 4. □