Performative Prediction with Bandit Feedback: Learning through Reparameterization

Yatong Chen [1] **Wei Tang** [2 3] **Chien-Ju Ho** [4] **Yang Liu** [1]

Abstract

Performative prediction, as introduced by Perdomo et al., is a framework for studying social prediction in which the data distribution itself changes in response to the deployment of a model. Existing work in this field usually hinges on three assumptions that are easily violated in practice: that the performative risk is convex over the deployed model, that the mapping from the model to the data distribution is known to the model designer in advance, and the first-order information of the performative risk is available. In this paper, we initiate the study of performative prediction problems that do not require these assumptions. Specifically, we develop a *reparameterization* framework that reparametrizes the performative prediction objective as a function of the induced data distribution. We then develop a two-level zeroth-order optimization procedure, where the first level performs iterative optimization on the distribution parameter space, and the second level learns the model that induces a particular target distribution at each iteration. Under mild conditions, this reparameterization allows us to transform the non-convex objective into a convex one and achieve provable regret guarantees. In particular, we provide a regret bound that is sublinear in the total number of performative samples taken and is only polynomial in the dimension of the model parameter.

[1]Department of Computer Science and Engineering, University of California, Santa Cruz, California, United States. [2]Data Science Institute, Columbia University [3]Department of Decisions, Operations, and Technology, the Chinese University of Hong Kong [4]Department of Computer Science and Engineering, Washington University in St. Louis. Correspondence to: Yang Liu <yangliu@ucsc.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Performative prediction, as introduced by Perdomo et al., provides a framework for studying prediction and risk minimization when the data distribution itself changes in response to the deployment of a model. Such phenomena, usually referred to as "performativity," are prevalent in various social prediction contexts, including education, recommendation systems, and criminal prediction, among others (Perdomo et al., 2020; Chen et al., 2023; Hardt et al., 2016; Dong et al., 2018; Kleinberg & Raghavan, 2020). For instance, consider a college admission process that places significant importance on standardized test scores. This process can incentivize students to invest more effort in test preparation, ultimately leading to a pool of applicants with much higher test scores than initially expected. This phenomenon is also prevalent in real-world applications, particularly in large-scale online recommendation systems, where the high frequency of updates to the recommendation algorithm can reshape users' future behavior. For example, video platforms such as TikTok, Netflix, and YouTube provide personalized recommendations that can influence users' future preferences and lead to shifts in the user-advertiser interaction patterns, thereby creating a dynamic and evolving data distribution.

More formally, consider the standard empirical risk minimization (ERM) problem defined by a loss function ℓ, a model parameter space $\Theta \subset \mathbb{R}^{d_\Theta}$ where $d_\Theta \in \mathbb{Z}_{>0}$, an instance space $Z = X \times Y$, and a *fixed* data distribution $\mathcal{D}$ over Z. The task is to find a model that minimizes the empirical risk defined as: $\mathsf{ER}(\theta, \mathcal{D}) := \mathbb{E}_{z\sim\mathcal{D}}[\ell(z;\theta)]$. Performative prediction extends this learning task by positing that the data distribution $\mathcal{D}$ is *not* fixed but is instead a function of the model parameter $\theta \in \Theta$. Here, we refer to $\mathcal{D}(\cdot)$ as a *distribution map*, and $\mathcal{D}(\theta)$ as the data distribution *induced* by the model θ. The objective is then to minimize the *performative risk*, defined as

$$\mathsf{PR}(\theta, \mathcal{D}(\theta)) := \mathbb{E}_{z\sim\mathcal{D}(\theta)}[\ell(z;\theta)] \,.$$

Intuitively, the performative prediction risk evaluates the performance of the model θ on the resulting distribution $\mathcal{D}(\theta)$ via the loss function ℓ. When it is clear from the context, we also use $\mathsf{PR}(\theta)$ to shorthand the performative

risk.

Optimizing the performative risk is generally challenging. In standard ERM, a convex loss function ℓ implies a convex empirical risk. However, as Perdomo et al. (2020) observed, the performative risk $\mathsf{PR}(\cdot)$ may be non-convex even when the loss ℓ itself is convex. For this reason, earlier works (Perdomo et al., 2020; Mendler-Dünner et al., 2020; Drusvyatskiy & Xiao, 2020; Brown et al., 2022) then focus on computing a *performative stable* solution instead, which is easier to achieve using standard optimization tools like repeated risk minimization. A performative stable model is loss-minimizing *on the data distribution it induces*, though other models may incur smaller losses on their respective induced distributions. However, as recent works (Miller et al., 2021; Izzo et al., 2021) point out, such stable solutions may be highly suboptimal and, worse yet, may not exist in certain settings.

One major challenge in performative risk minimization is the unknown distribution map between the model parameter θ and the distribution $\mathcal{D}(\theta)$ without making any structural assumption. For example, one can hardly anticipate the click-through rate of an ad without putting out the ad first. In the language of performative prediction, only by deploying a model θ can the learner observe data samples that are i.i.d realized from the induced data distribution $\mathcal{D}(\theta)$. Due to this inherent uncertainty about $\mathcal{D}(\theta)$, it is impossible to compute the gradient of $\mathsf{PR}(\theta)$ w.r.t θ, not to mention finding a model with the lowest performative risk offline. Instead, the learner must interact with the environment and deploy models θ to explore the induced distributions $\mathcal{D}(\theta)$, which involves deploying "imperfect" models on decision subjects.

In this paper, we propose to measure the loss incurred by deploying a sequence of models $\theta_1, \dots, \theta_{T_{\mathsf{total}}}$ by evaluating the following regret measured with respect to the total number of samples deployed during the process:

$$\mathcal{R}_N(\mathcal{A}, \mathsf{PR}) = \sum_{\tau=1}^{T_{\mathsf{total}}} \sum_{i=1}^{n_\tau} \ell(z_\tau^{(i)}; \theta_\tau) - N \cdot \mathsf{PR}(\theta_{\mathtt{OPT}}) \quad (1)$$

where $N := \sum_{\tau=1}^{T_{\mathsf{total}}} n_\tau$ denotes the total number of observed data samples throughout the process, $\mathcal{A}$ corresponds to the particular algorithm, and PR represents the objective function. This regret measures the suboptimality of the deployed sequence of models relative to a performative optimum $\theta_{\mathtt{OPT}} \in \operatorname{argmin}_\theta \mathrm{PR}(\theta)$ in terms of how much loss they incur on the population with N decision subjects.

In contrast to earlier studies that primarily assess the final model's performance based on optimality rather than the cumulative loss incurred throughout the process, we argue that this constitutes a more practical evaluation metric in predictive scenarios involving multiple rounds of human feedback. In particular, since the process of finding the optimal performative model involves deploying sub-optimal models on human agents in the process, it is more appropriate to define regret on the total number of agents that are subjected to the "imperfect" algorithmic system rather than only caring about whether the final model is optimal. We believe this provides a unique evaluation metric suitable for performative prediction.

Later in Section 4.3, we compare our proposed regret definition with the standard regret measured in T_{total} in more detail and show that our algorithm is, in fact, also sublinear in the total deployment steps T_{total}. This, combined with the fact that sublinear regret implies model convergence (Proposition 1), also means that our algorithm can guarantee to output a model arbitrarily close to the performative optimal model $\theta_{\mathtt{OPT}}$.

1.1. Our Contributions

Our main contributions are a two-level zeroth order optimization algorithm that achieves a sublinear regret bound measured using the total number of samples and a novel reparametrization framework attempting to tackle a particular non-convex performative prediction problem.

Reparametrization Framework. Departing from previous work, we allow $\mathsf{PR}(\theta)$ to be non-convex in the model parameter θ, but suppose it is convex in the *data distribution* parameter $\phi \equiv \varphi(\theta)$. Informally, under mild conditions, we show that non-convex $\mathsf{PR}(\theta)$ can be reparameterized as a new (convex) function $\mathsf{PR}^\dagger(\phi)$ over the induced data distribution parameter ϕ. We discuss detailed parametrization procedure in Section 3.

Zeroth-Order Optimization Algorithm with Performativity. Given the parametrization framework proposed above, we propose a two-level zeroth-order optimization procedure, which, to our knowledge, is novel in performative prediction. We believe our method enjoys the following benefits:

- **No Requirement for Gradient Information** Unlike the traditional gradient-based optimization procedure, our method does not require the explicit calculation of *gradients* that may be complex or unavailable.
- **Black-Box Models** Our method can still be effective when dealing with models or systems that are treated as black boxes, where the internal mechanisms are not well understood (such as complicated economic systems) since it doesn't require knowledge of the underlying model structure.
- **Robustness to Noise** In many real-world applications, objective function evaluations may be noisy or subject to uncertainty, such as modeling consumer behavior. Our method can handle noisy evaluations and make decisions that are robust to noise.

Our main results can be summarized as follows:

Theorem 1 (Informal)**.** *There exists an algorithm that, under appropriate conditions, incurs regret* $\widetilde{O}((d_\Theta + d_\Phi) \cdot N_{\mathsf{KL}}^{1/6} \cdot N^{5/6})$[1] *after* N *performative samples*[2] *with probability at least* $1 - p$*, where* N_{KL} *depends on the sample efficiency of an off-the-shelf estimator for KL divergence, and* d_Θ *and* d_Φ *denote the dimension of the model and distribution parameter space, respectively.*

The N_{KL} term in our regret depends on the sample efficiency of the estimator for KL divergence. The detailed discussion is provided in Section 4.2.

1.2. Related Work

Our work most closely relates to performative prediction and zeroth-order optimization. Due to page limit, we include additional related work in Appendix G, including detailed comparisons of our work to three closely related jobs (Jagadeesan et al., 2022; Miller et al., 2021; Maheshwari et al., 2022), and more recent developments of performative prediction.

Performative Prediction. Performative Prediction, first explored in Perdomo et al. (2020), has recently received many follow-up works, including but not limited to Miller et al. (2021); Izzo et al. (2021); Drusvyatskiy & Xiao (2020); Mendler-Dünner et al. (2020); Brown et al. (2022); Jagadeesan et al. (2022); Dong & Ratliff (2021); Cutler et al. (2021) and Piliouras & Yu (2022). These works mostly focus on the performative stability and the performative optimality, including developing an algorithmic procedure that converges to performatively stable or optimal points. Similar to this line of research (Dong & Ratliff, 2021; Jagadeesan et al., 2022; Izzo et al., 2021; Miller et al., 2021), our work also focuses on performative optimality.

Zeroth-Order Optimization. Our algorithms and techniques are based on the line of work on zeroth-order optimization (also known as bandit optimization) initiated by Flaxman et al. (2005), which studies how to optimize an unknown convex function f using only function value query access to f. Agarwal et al. (2010) and Shamir (2017) later extend the technique that allows multiple points query and show that two points suffice to guarantee that the regret bounds that closely resemble the regret bounds for the full information case. The reparameterization approach proposed in our paper mirrors the intuition behind the algorithms proposed for learning from revealed feedback or preferences (see, e.g., Roth et al. (2016); Zadimoghaddam & Roth (2012); Dong et al. (2018)), which consider a *Stackelberg game* involving a utility-maximizing learner and strategic agent. Our work, focusing on performative prediction with an environment response exogenously characterized by a distribution map $\mathcal{D}(\cdot)$, differs from theirs in problem consideration.

[1] $\widetilde{O}(\cdot)$ suppresses polylogarithmic factors in N and the failure probability $1/p$.

[2] Samples that the learner deploys along the way of finding the performative optimal model.

1.3. Key Notations

Let $d_\Theta \in \mathbb{Z}_{>0}$ denote the dimension of the model parameter θ, and let $D_\Theta := \sup\{\|\theta - \theta'\|, \forall \theta, \theta' \in \Theta\}$ denote the diameter of the model parameter space Θ. The data distribution $\mathcal{D}(\theta)$ has a parametric continuously differentiable density $p(z; \varphi(\theta))$ where $\varphi(\theta)$ denote the distribution parameter for $\mathcal{D}(\theta)$. We use $\varphi(\cdot)$ to denote the distribution parameter mapping while ϕ to denote a given distribution parameter. Let $d_\Phi \in \mathbb{Z}_{>0}$ denote the dimension of the model parameter ϕ, and let $D_\Phi := \sup\{\|\phi - \phi'\| \,|, \forall \phi, \phi' \in \Phi\}$ denote the diameter of the model parameter space Φ. When it is clear from the content, we use $\varphi(\theta)$ to represent $\mathcal{D}(\theta)$ the distribution θ induces. Let $\vartheta^*(\phi)$ denote the *optimal* model parameter that induces a specific target distribution parameter ϕ – in case of having multiple model parameters that potentially induce the same distribution parameter ϕ, $\vartheta^*(\phi)$ is the one that achieves the minimum performative prediction risk.

1.4. Structure of the Paper

The rest of the paper is organized as follows: In Section 2, we introduce the problem formulation and provide a warm-up setting when $\mathsf{PR}(\theta)$ is convex over the model parameter θ. Using this simple setting, we introduce the zeroth-order optimization technique we use, which will serve as the building block to solve for a more complicated setting (i.e., when $\mathsf{PR}(\theta)$ is *not* convex over θ). We also present a fundamental fact in convex optimization that sublinear regret implies model convergence (Proposition 1), which unifies the goal of regret minimization and model optimality in our setting. In Section 3, we provide an overview of our proposed solution. In Section 4.1, we describe the outer algorithm, and Section 4.2 describes the inner algorithm called $\mathsf{LearnModel}$, which is used to solve a subroutine problem using black-box oracle. Section 4.3 contains the overall regret analysis. Lastly, in Section 5, we discuss the limitations and potential future work. *All omitted proofs can be found in the Appendix.*

2. Preliminaries

We begin by formally defining our problem.

2.1. Problem Formulation

The objective of performative prediction is to minimize the *performative risk* defined as $\mathsf{PR}(\theta) := \mathbb{E}_{z \sim \mathcal{D}(\theta)}[\ell(z; \theta)]$.

A model $\theta_{\texttt{OPT}} \in \Theta$ is said to be *performatively optimal* if $\mathsf{PR}(\theta_{\texttt{OPT}}) = \min_{\theta\in\Theta} \mathsf{PR}(\theta)$. To find the performatively optimal model, one usually needs to have the full knowledge of the underlying distribution map $\mathcal{D} : \Theta \to \Phi$. In this work, we consider a more practical scenario where the distribution map $\mathcal{D}$ is not known in advance, and to learn the performatively optimal model, the learner has to adaptively deploy models to gradually learn the underlying distribution map.

Formally, we consider the following repeated interaction between the learner and the environment consisting of decision subjects where we can only query through samples. The interaction proceeds for T_{total} steps, at each time step $\tau = 1, \ldots, T_{\mathsf{total}}$: (1) the learner deploys a model $\theta_\tau \in \Theta$; (2) the learner observes n_τ data samples $\{z_\tau^{(i)}\}_{i\in[n_\tau]}$ where each $z_\tau^{(i)} \overset{\text{iid}}{\sim} \mathcal{D}(\theta_\tau)$; (3) the learner incurs empirical loss $\ell(z_\tau^{(i)}; \theta_\tau)$ for each sample.

The goal of the learner is to design an online model deployment policy $\mathcal{A}$ such that it minimizes her cumulative empirical risk over all observed data samples:

$$\mathcal{R}_N(\mathcal{A}, \mathsf{PR}) = \sum_{\tau=1}^{T_{\mathsf{total}}} \sum_{i=1}^{n_\tau} \ell(z_\tau^{(i)}; \theta_\tau) - N \cdot \mathsf{PR}(\theta_{\texttt{OPT}}) \quad (2)$$

where $N := \sum_{\tau=1}^{T_{\mathsf{total}}} n_\tau$ denotes the total number of observed data samples throughout the process. The reason we introduce T_{total} instead of N directly is that each step (τ) of our algorithm performs different tasks, where we would impose different requirements of samples to be collected. This shall become clear later when we present our algorithm in the following sections.

2.2. Warmup Setting: When $\mathsf{PR}(\theta)$ is Convex in θ

In this section, we analyze a simple scenario when we assume that the performative risk $\mathsf{PR}(\theta)$ is convex over the model parameter θ. The technique we use to solve this simple case will be the building block to solve the later more challenging problem where $\mathsf{PR}(\theta)$ is *not* convex over the model parameter θ.

Recall that when the learner deploys a model θ, she observes a set of data samples which are i.i.d drawn from the underlying data distribution $\mathcal{D}(\theta)$. This enables us to compute an unbiased estimate $\widetilde{\mathsf{PR}}(\theta)$ for the performative risk $\mathsf{PR}(\theta)$ of the deployed model θ:

$$\widetilde{\mathsf{PR}}(\theta) = \frac{1}{n_\tau} \sum_{i=1}^{n_\tau} \ell(z_\tau^{(i)}; \theta), \text{ and } \mathbb{E}[\widetilde{\mathsf{PR}}(\theta)] = \mathsf{PR}(\theta), \forall \theta \in \Theta$$

where the expectation is over the randomness of the observed samples. Since $\mathsf{PR}(\theta)$ is convex over the model parameter θ, one can use an off-the-shelf zeroth-order convex optimization technique (Agarwal et al., 2010) to solve this problem and get the following regret guarantee:

Lemma 1. *When $\mathsf{PR}(\theta)$ is convex, L-Lipschitz w.r.t. the deployed model parameter θ, there exists an algorithm (Algorithm 3) achieving $\mathcal{R}_N(\mathcal{A}_3, \mathsf{PR}) = O(\sqrt{d_\Theta N \log \frac{1}{p}})$ with probability at least $1 - p$, where N is the total number of samples deployed during the process.*

We defer the proof and the details of Algorithm 3 to Appendix B. In particular, Algorithm 3 deploys two models at each time step, in doing so, one can show that the regret bounds closely resemble bounds for the full information case where the learner knows the distribution map $\mathcal{D}(\cdot)$. The proof of the above result builds on the main result of Agarwal et al. (2010), and also incorporates an improved analysis of the gradient estimate due to Shamir (2017).

2.3. Useful Fact: Sublinear Regret Implies Convergence in Model Optimality

A folklore fact in online and zeroth-order optimization is that if a function f is convex and we wish to converge to an approximately optimal point, it suffices to show a query algorithm that achieves $o(n)$ regret after n queries. In particular, we have the following proposition:

Proposition 1 (Sublinear Regret Implies Convergence)**.** *Let $f : X \to \mathbb{R}$ be convex, and let $\mathcal{A}$ be an algorithm for minimizing f whose regret after n queries is sublinear in n, i.e. $\mathcal{R}_n(\mathcal{A}, f) = o(n)$. Then we can compute an ϵ-suboptimal point for f in $\mathcal{R}_n(\mathcal{A}, f)/\epsilon$ queries of f.*

This proposition establishes a strong link between achieving sublinear regret and the convergence toward an optimal model. It implies that if our proposed algorithm attains a sublinear regret as defined in Equation 1, this automatically suggests that we can obtain an almost optimal model, which is exceptionally close to the truly optimal model, denoted as $\theta_{\texttt{OPT}}$. This closeness is achieved simply by averaging the models $\theta_1, \ldots, \theta_{T_{\mathsf{total}}}$ throughout the deployment process. This helps us unify the goal of regret minimization and finding the optimal model.

3. Optimizing PR via Reparameterization: An Overview of Our Solution

When $\mathsf{PR}(\theta)$ is not convex over the model parameter θ, the zeroth-order convex optimization technique used in Section 2.2 is not directly applicable. Instead, we leverage the structure of $\mathsf{PR}(\theta)$ and *reparameterize* it as a function of the *induced* data distribution $\mathcal{D}(\theta)$. In particular, we consider the setting where the data distribution $\mathcal{D}(\theta)$ has a parametric continuously differentiable density $p(z; \varphi(\theta))$, and the functional form $p(z; \phi)$ is known to the learner but the distribution parameter ϕ remains unknown. Under mild

conditions, we show that the performative risk $\mathsf{PR}(\theta)$ can be reformulated as a function of the *induced* distribution distribution parameter $\phi \equiv \varphi(\theta)$, namely,

$$\mathsf{PR}(\theta) = \mathsf{PR}^\dagger(\varphi(\theta)) \equiv \mathsf{PR}(\vartheta^*(\phi)) \,, \tag{3}$$

and $\mathsf{PR}^\dagger(\phi)$ is convex over the distribution parameter ϕ (See more details in Section 4.1).

Here we provide two real-life settings to justify our model:

Example 1. *(Biased coin flip). Consider the task of predicting the outcome of a biased coin flip similar to Perdomo et al. (2020), where the bias of the coin depends on a feature X and the assigned score $f_\theta(X)$. In particular, define $\mathcal{D}(\theta)$ in the following way: X is a 1-dimensional feature supported on $[0,1]$ and $Y \sim \text{Bernoulli}(\varphi(\theta))$. Assume that the class of predictors consists of linear models of the form $f_\theta(x) = \theta x$ and that the objective is to minimize the squared loss: $\ell(x, y; \theta) = (y - f_\theta(x))^2$. When the probability of the coin landing on heads $\varphi(\theta) = \theta^2$, we can verify that $\mathsf{PR}(\theta)$ is convex in $\varphi(\theta)$, not in θ (by similar argument provided below in Example 3).*

Example 2. *(Expected revenue of goods). Let $\theta \in \mathbb{R}^d$ denote a vector of prices for various goods the distributor sets. A vector z denotes a customer's demand for each good. The distributor's goal is to maximize the expected revenue $\mathsf{PR}(\theta) = \mathbb{E}_{z\sim\mathcal{D}(\theta)}\left[\theta^\top z\right]$. In other words, the loss function is $\ell(z;\theta) = -\theta^\top z$. When $\mathcal{D}(\theta) = N(\varphi(\theta), \Sigma^2)$ with $\varphi(\theta) = \sqrt{\theta}$ and a fixed Σ^2, we can verify that $\mathsf{PR}(\theta)$ is not convex in θ but is convex in $\varphi(\theta) = \sqrt{\theta}$.*

With this reparameterization, one can operate on the space of distribution parameters and hopefully apply the zeroth-order convex optimization technique. However, one notable challenge is in zeroth-order convex optimization, the learner is usually assumed to have direct query access to the unknown convex function f. Namely, when querying point x, the learner is able to immediately obtain the information about the (noisy) value $f(x)$. In our setting, such direct access is, unfortunately, not available since the mapping $\varphi(\cdot)$ is not known to the learner. Indeed, the learner can only deploy a model θ to observe the empirical performative risk $\widetilde{\mathsf{PR}}(\theta)$ which is evaluated over the observed data samples drawn from the induced data distribution $\mathcal{D}(\theta)$. Hence, to evaluate the value $\mathsf{PR}^\dagger(\phi)$ on a target data distribution with the parameter ϕ, we use another algorithm called $\mathsf{LearnModel}$ as a subroutine to find a model $\bar{\theta}$ such that $\varphi(\bar{\theta}) \approx \phi$ (See Section 4.2).

Summary of our proposed procedure. Intuitively, the outer loop optimizes the objective function PR in the distribution parameter space $\phi \in \Phi$ iteratively and tries to find the optimal data parameter $\phi*$, while the inner loop ($\mathsf{LearnModel}$) tries to find a model parameter to induce the particular data parameter that the outer loop is currently iterating on.[3] A graph illustration of our algorithm procedure in given in Figure 1.

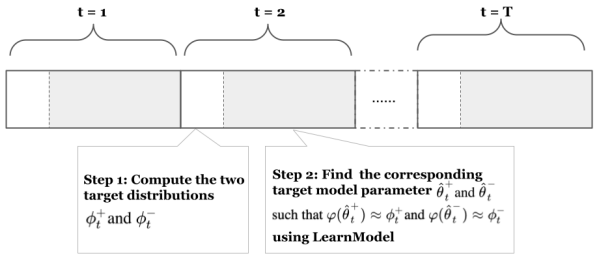

Figure 1. Illustration of our procedure (Algorithm 1). Each big block represents one iteration of the outer algorithm, which consists of three sub-steps: Step 1, the learner first computes the two target distribution ϕ_t^+ and ϕ_t^- (corresponds to the white section), Step 2, the learner uses $\mathsf{LearnModel}$ to learn the corresponding model $\hat{\theta}_t^+$ and $\hat{\theta}_t^-$ that can best approximately induce ϕ_t^+ and ϕ_t^- (corresponds to the grey section) correspondingly. Step 3, the learner deploys $\hat{\theta}_t^+$ and $\hat{\theta}_t^-$ and perform a gradient update and get ϕ_{t+1}. Each deployment of $\mathsf{LearnModel}$ requires a total number of S steps. Thus, the total number of steps involved in the whole procedure is $T_{\text{total}} = T \times S$.

3.1. Examples of PR Being Convex in ϕ not in θ

We first provide three examples in which our condition (PR loss is convex in the induced distribution parameter $\varphi_\theta := \phi(\theta)$, but non-convex in the model parameter θ) holds. See more examples and derivation details in Appendix C.

Example 3. *(Bernoulli distribution) Consider the following one-dimension linear model with the squared loss $\ell(\theta; (x, y)) = -(\theta x - y)^2$. Assuming a model $\theta \in \Theta = [0, 1]$ induces a Bernoulli distribution over the labels with the distribution parameter $\varphi(\theta) := \theta^2$, i.e., $y \sim \mathsf{Bern}(\varphi(\theta))$. Then its PR loss is convex in its data parameter φ_θ but not convex in its model parameter θ.*

Notice that the example provided can be generalized to any distribution map $\phi(\theta)$ that satisfies $\varphi(\theta) = \theta^\alpha$ for any $\alpha > 1$, and any ℓ_β loss for even β value. In addition, Example 1 can also be any generalized monotone polynomial function $\varphi(\theta)$.

[3] One may wonder how to find the optimal $\vartheta^*(\phi)$ when there are two model parameters θ and θ' that realize the same ϕ (i.e., $\varphi(\theta) = \varphi(\theta')$ and $\mathsf{PR}(\theta') \geq \mathsf{PR}(\theta)$). Recall that the objective function for $\mathsf{LearnModel}$ is to find *any* model θ that leads to the particular target data parameter ϕ such that $\varphi(\theta) = \phi$. It is quite possible that multiple models can induce the same target data parameter; however, since the goal is to find any one of them, having multiple model parameters won't be an issue – in fact, it can only help speed up the process

Example 4 (Gaussian distribution). *For a random variable x following a one-dimensional Gaussian distribution with fixed variance, i.e., $\mathcal{D}(\theta) = N(\varphi(\theta), \sigma^2)$, and let the loss as $\ell(x;\theta) = (\theta x)^2$, then we have $\mathsf{PR}(\theta) = \theta^2(\sigma^2 + \varphi(\theta)^2)$. With $\varphi(\theta) = \sqrt{\theta}$, we verify that $\mathsf{PR}(\theta)$ is not convex in the model parameter θ but is convex in the distribution parameter $\varphi(\theta) = \sqrt{\theta}$.*

Example 5 (Uniform distribution). *Fix a random variable x. Let y follow a uniform distribution with parameter $\varphi(\theta)$, e.g., $y \sim$ Uniform$[0, \varphi(\theta)]$, and with the loss being $\ell(\theta; x, y) = -(\theta x - y)^2$, we have $\mathsf{PR}(\theta) = \theta^2 x^2 - \theta x \varphi(\theta) + \frac{1}{3}\varphi(\theta)^2$. Setting $\varphi(\theta) = \theta^2$, we verify that it's convex in the distribution parameter $\varphi(\theta)$ but not in the model parameter θ.*

Remark 1. Earlier work (Miller et al., 2021) posits the "mixture dominance assumption", under which the performative prediction risk turns out to be convex in the model parameter θ. In particular, the assumption requires that for any triple $\theta, \theta', \theta_0 \in \Theta$, the following condition holds: $\mathbb{E}_{z\sim\mathcal{D}(\alpha\theta+(1-\alpha)\theta')}[\ell(\theta_0; z)] \leq \mathbb{E}_{z\sim\alpha\mathcal{D}(\theta)+(1-\alpha)\mathcal{D}(\theta')}[\ell(\theta_0; z)]$. The primary distinction between our condition and theirs is that our condition only needs to be valid for each individual data parameter. This is in contrast to Miller et al.'s condition, which must be met for any combination of θ, θ', and $\theta_0 \in \Theta$. We believe our approach has greater versatility and is more likely to be fulfilled in various scenarios [4].

4. Our Algorithm and its Performance Guarantee

In this section, we provide the details of our proposed algorithm, and also the associated performance analysis.

4.1. The Outer Algorithm: A Reparameterization Approach

As we mentioned, in this work, we study the scenario where $\mathsf{PR}(\theta)$ is not convex over the model parameter. The high-level idea is that we can *reparameterize* the performative risk $\mathsf{PR}(\theta)$ as a function $\mathsf{PR}^\dagger(\phi)$ over the data distribution parameter ϕ. In particular, we first reformulate the learner's loss function so that it can be expressed as a function *only* in the induced data distribution. For each data distribution $\phi \in \Phi$, assume the set of learner's actions (deployed model parameters) that induce ϕ is $\Theta^*(\phi) = \{\theta \in \Theta | \varphi(\theta) = \phi\}$ Among all of the learner's actions that induce ϕ, the optimal one that achieves the minimal PR loss across the whole population is:

$$\vartheta^*(\phi) = \operatorname*{argmin}_{\theta\in\Theta^*(\phi)} \mathsf{PR}(\theta)$$

where ties are broken arbitrarily. Now we can rewrite learner's objective function as a function of ϕ

$$\mathsf{PR}^\dagger(\phi) = \mathsf{PR}(\vartheta^*(\phi)) \tag{4}$$

To make the problem tractable, we consider following generic class of $\mathsf{PR}^\dagger(\cdot)$ that is convex and Lipchitz continuous.

Assumption 1. *$\mathsf{PR}^\dagger(\phi)$ is convex and $L^\dagger$-Lipschitz over the data distribution parameter $\phi \in \Phi$.*

With reparameterizing $\mathsf{PR}(\theta)$ as a function $\mathsf{PR}^\dagger(\phi)$ over the induced data distribution parameter ϕ, we now wish to minimize a bounded, $L^\dagger$-Lipschitz function $\mathsf{PR}^\dagger(\cdot) : \Phi \to \mathbb{R}$, where $\Phi \subset \mathbb{R}^{d_\Phi}$ has bounded diameter D_Φ, by operating on the distribution parameter space Φ.

Instead of having immediate query access in zeroth-order convex optimization algorithm, in our setting, we cannot directly evaluate the (noisy) value $\mathsf{PR}^\dagger(\phi)$ for a particular data distribution parameter, but may query the following oracles:

- A noisy *function oracle* $\mathsf{EstimatePR}$, which takes $\theta \in \Theta$ as input and returns an unbiased estimate $\widetilde{\mathsf{PR}}$ such that $\mathbb{E}[\widetilde{\mathsf{PR}}(\theta)] = \mathsf{PR}(\theta)$. This noisy oracle can be implemented by simply making a prediction and observing the loss as defined in Section 2.2 [5].
- A noisy *reparameterization oracle* $\mathsf{LearnModel}(\phi, \epsilon_{\mathsf{LM}}, p_{\mathsf{LM}})$, which takes $\phi \in \Phi$, $\epsilon_{\mathsf{LM}}, p_{\mathsf{LM}} > 0$ as input and returns $\theta \in \Theta$ such that $\Pr(\|\varphi(\theta) - \phi\| \geq \epsilon_{\mathsf{LM}}) \leq p_{\mathsf{LM}}$. We will specify $\mathsf{LearnModel}$ in Section 4.2.

Algorithm 1 achieves this task. Specifically, it returns both $\bar{\theta} \in \Theta$ and $\bar{\phi} \in \Phi$ such that with probability at least $1 - p$, $|\mathsf{PR}(\bar{\theta}) - \mathsf{PR}(\theta_{\mathtt{OPT}})| \leq \epsilon$ and $|\mathsf{PR}^\dagger(\bar{\phi}) - \mathsf{PR}^\dagger(\varphi(\theta_{\mathtt{OPT}}))| \leq \epsilon$.

For analysis purpose, we also define regret in T, the total number of steps $\mathsf{MinimizePR}$ has to go through in order to get an ϵ-suboptimal model parameter w.r.t the PR objective function:

$$\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR}) = \sum_{t=1}^{T} \left[\mathsf{EstimatePR}(\hat{\theta}_t^+) + \mathsf{EstimatePR}(\hat{\theta}_t^-) - 2\mathsf{PR}(\theta_{\mathtt{OPT}})\right]$$

We demonstrate the following regret bound for this algorithm:

Theorem 2 (High-probability regret bound for Algorithm 1 in T). *When Algorithm 1 is called with arguments ϵ_{LM} and*

[4] We can also verify that Example 3 does not satisfy the mixture dominance assumption.

[5] The sample required for each round of estimate EstimatePR is $\mathcal{O}(1)$; this is because EstimatePR itself is an unbiased estimator, so even with one sample, in expectation, the estimation will be unbiased.

Algorithm 1 Bandit algorithm for minimizing an indirectly convex function with noisy oracles

function EstimatePR(θ) ▷ Unbiased estimate of PR(θ)
  Deploy θ, observe sample $z \sim \mathcal{D}(\theta)$
  return $\ell(z;\theta)$
function MinimizePR(LearnModel : $\Phi \to \Theta$; $\epsilon, p, \epsilon_{\mathsf{LM}}, p_{\mathsf{LM}} > 0$)
  $T \leftarrow \frac{d_\Phi}{(\epsilon - \sqrt{\epsilon_{\mathsf{LM}} d_\Phi})^2}$, $\delta \leftarrow \sqrt{\epsilon_{\mathsf{LM}} d_\Phi}$, $\eta \leftarrow 1/\sqrt{d_\Phi T}$
  $y_1 \leftarrow \mathbf{0}$
  for $t \leftarrow 1, \dots, T$ **do**
    $u_t \leftarrow$ sample from $\mathrm{Unif}(\mathbb{S})$
    $\phi_t^+ \leftarrow \phi_t + \delta u_t$, $\phi_t^- \leftarrow \phi_t - \delta u_t$
    $\hat{\theta}_t^+ \leftarrow \mathsf{LearnModel}(\phi_t^+, \epsilon_{\mathsf{LM}}, p_{\mathsf{LM}})$
    $\hat{\theta}_t^- \leftarrow \mathsf{LearnModel}(\phi_t^-, \epsilon_{\mathsf{LM}}, p_{\mathsf{LM}})$ ▷ $\hat{\theta}_t^+$ such that $\mathsf{PR}(\hat{\theta}_t^+) \approx \mathsf{PR}^\dagger(\phi_t^+)$, similarly, $\hat{\theta}_t^-$ such that $\mathsf{PR}(\hat{\theta}_t^-) \approx \mathsf{PR}^\dagger(\phi_t^-)$
    $\widetilde{\mathsf{PR}}(\hat{\theta}_t^+) \leftarrow \mathsf{EstimatePR}(\hat{\theta}_t^+)$
    $\widetilde{\mathsf{PR}}(\hat{\theta}_t^-) \leftarrow \mathsf{EstimatePR}(\hat{\theta}_t^-)$ ▷ Approximations of $\mathsf{PR}(\hat{\theta}_t^+)$, $\mathsf{PR}(\hat{\theta}_t^-)$
    $\tilde{g}_t \leftarrow \frac{d_\Phi}{2\delta}\left(\widetilde{\mathsf{PR}}(\hat{\theta}_t^+) - \widetilde{\mathsf{PR}}(\hat{\theta}_t^-)\right) \cdot u_t$ ▷ Approximation of $\nabla_\phi \mathsf{PR}^\dagger(\phi_t)$
    $\phi_{t+1} \leftarrow \Pi_{(1-\delta)\Phi}(\phi_t - \eta \tilde{g}_t)$ ▷ Take gradient step and project
  $\bar{\phi} \leftarrow \frac{1}{T}\sum_{t=1}^T \phi_t$
  $\bar{\theta} \leftarrow \mathsf{LearnModel}(\bar{\phi}, \epsilon_{\mathsf{LM}}, p_{\mathsf{LM}})$
  return $\bar{\theta}, \bar{\phi}$

p_{LM}, *we have for every* $p > 0$ *that*

$$\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR}) = O\left(\sqrt{d_\Phi T} + \sqrt{\epsilon_{\mathsf{LM}} d_\Phi} \cdot T + \sqrt{T \log \frac{1}{p}}\right)$$

with probability at least $1 - p - 2Tp_{\mathsf{LM}}$.

The above Theorem 2 requires that the output of LearnModel is ϵ_{LM}-close to the target distribution parameter ϕ with probability at least $1 - p_{\mathsf{LM}}$. Later in Section 4.2, we show how we achieve this by developing an zeroth-order convex optimization algorithm with the objective of minimizing the KL divergence of two distributions.

4.2. Inner Algorithm: Inducing a Target Distribution Using LearnModel

In this section, we show how to solve the sub-problem LearnModel mentioned in Algorithm 1: given a target distribution with the parameter $\phi \in \Phi$, find a model $\theta \in \Theta$ whose corresponding distribution parameter $\varphi(\theta)$ is close to ϕ.

Objective function for LearnModel. To this end, we consider minimizing the KL divergence between ϕ and $\varphi(\theta)$: [6]

$$\mathsf{KL}(\phi||\varphi(\theta)) := \int_z p(z;\phi) \log \frac{p(z;\phi)}{p(z;\varphi(\theta))} dz \quad (5)$$

where $p(z;\phi)$ denotes the pdf for the target distribution ϕ, and $p(z;\varphi(\theta))$ denotes the pdf for the distribution induced by deploying θ.

In general, $\mathsf{KL}(\phi||\varphi(\theta))$ measures how much a distribution with the parameter $\varphi(\theta)$ is away from the target distribution with the parameter ϕ: if the two distributions $\phi_1, \phi_2 \in \Phi$ satisfy $\phi_1 = \phi_2$, then $\mathsf{KL}(\phi_1||\phi_2) = 0$, otherwise $\mathsf{KL}(\phi_1||\phi_2) > 0$. Intuitively, the lower the value $\mathsf{KL}(\phi_1||\phi_2)$ is, the better we have matched the target distribution with our approximate distribution induced by the chosen model. However, $\mathsf{KL}(\phi||\cdot)$ is generally not convex nor Lipschitz. Hence, to make the problem tractable, we will make several assumptions. We view these assumptions as comparatively mild, and provide examples shortly after stating the assumptions we need.

Assumption 2. *The function* $\mathsf{KL}(\phi||\varphi(\cdot))$*, the data distribution* $\mathcal{D}(\theta)$*, and its parameter mapping* $\varphi(\cdot)$ *satisfies the following properties.*

- 2a. $\mathsf{KL}(\phi||\varphi(\cdot))$ *is convex in the model parameter* $\theta \in \Theta$*;*
- 2b. *The data distribution* $\mathcal{D}(\theta)$ *with the parameter* $\varphi(\theta)$ *is* (ℓ_2, K)*-Lipschitz continuous in the model parameter* $\theta \in \Theta$ *with constant* $K(z), \forall z \in Z$ [7]*;*
- 2c. *Let* $\mathcal{D}_1, \mathcal{D}_2$ *be two data distributions with the parameter* $\phi_1, \phi_2 \in \Phi$*, and* $d_{\mathsf{TV}}(\mathcal{D}_1, \mathcal{D}_2)$ *be the total variation distance. Then* $\|\phi_1 - \phi_2\| \le L_{\mathsf{TV}} \cdot d_{\mathsf{TV}}(\mathcal{D}_1, \mathcal{D}_2)$ *for some constant* $L_{\mathsf{TV}} > 0$*.*

Here, we provide examples to demonstrate that the above assumptions are comparatively mild. The following is an example showing the convexity of $\mathsf{KL}(\phi||\varphi(\cdot))$.

Example 1. *Consider the density function* $p(z;\varphi(\theta))$ *of the data distribution* $\mathcal{D}(\theta)$ *satisfying* $p(z;\varphi(\theta)) = \mathrm{Unif}(\exp(c\varphi(\theta)))$ *for some constant* $c > 0$ *and for any convex function* $\varphi(\theta)$*, then* $\mathsf{KL}(\phi||\varphi(\cdot))$ *is convex over* θ*.*

In the above Assumption 2b, we assume a family of distribution called the (ℓ_2, K)-Lipschitz continuous. This Lipschitz continuity over the parametrization of probability distributions allows us to have the following Lipschitz condition of the function $\mathsf{KL}(\phi||\varphi(\cdot))$ over the model parameter θ:

[6] For notation simplicity, here, we use $\mathsf{KL}(\phi_1||\phi_2)$ to represent $\mathsf{KL}(\mathcal{D}_1||\mathcal{D}_2)$ where the data distribution $\mathcal{D}_1$ and $\mathcal{D}_2$ has the parameter ϕ_1 and ϕ_2, respectively.

[7] A distribution $\mathcal{D}(\theta)$ with the density function $p(\cdot|\varphi(\theta))$ parameterized by $\theta \in \Theta$ is called (ℓ_2, K)-Lipschitz continuous (Honorio, 2011) if for all z in the sample space, the log-likelihood $f(\theta) = \log p(z|\varphi(\theta))$ is Lipschitz continuous with respect to the ℓ_2 norm of θ with constant $K(z)$.

Lemma 2 (Lipschitzness of $\mathsf{KL}(\phi||\varphi(\theta))$ in θ)**.** *Given two (ℓ_2, K)-Lipschitz continuous distributions $\mathcal{D}_1 = p(\cdot \mid \varphi(\theta_1))$ and $\mathcal{D}_2 = p(\cdot \mid \varphi(\theta_2))$, and a target distribution parameter $\phi \in \Phi$, we have $|\mathsf{KL}(\phi||\varphi(\theta_1)) - \mathsf{KL}(\phi||\varphi(\theta_2))| \leq L_{\mathsf{KL}} \|\theta_1 - \theta_2\|$ with a constant $L_{\mathsf{KL}} > 0$.*

The above Assumption 2c is about the continuity on the distribution parameter $\phi \in \Phi$. Intuitively, this assumption ensures that if the parameters of two distribution are close, then their total variation distance is close as well. With this assumption, we can show that the distance between two distribution parameters $\|\phi_1 - \phi_2\|$ can be bounded by the KL divergence between the corresponding data distributions.

Lemma 3. *With Assumption 2c, we have $\|\phi_1 - \phi_2\| \leq L_\phi \sqrt{\mathsf{KL}(\phi_1||\phi_2)}$ for some constant $L_\phi > 0$.*

Intuitively, the above result ensures that given a target distribution parameter ϕ, as long as a model θ whose corresponding data distribution is close (i.e., $\mathsf{KL}(\phi||\varphi(\theta))$ is small) to the distribution with the parameter ϕ, then $\varphi(\theta)$ is close to ϕ. We will use Lemma 3 in the proof of our main theorem in Section 4.3.

Algorithm for LearnModel**.** When $\mathsf{KL}(\phi||\varphi(\cdot))$ is convex and Lipschitz over the model θ, its minimizer can be computed using algorithms similar to Algorithm 1. In our problem, given a target data distribution with the parameter ϕ, we can use the observed data samples to approximately compute the $\mathsf{KL}(\phi||\varphi(\theta))$ when deploying a model θ. Indeed, we assume an existence of an oracle $\mathsf{EstimateKL}(\phi, (z_t^{(i)})_{i\in[n_t]})$ which takes the observed samples $(z_t^{(i)})_{i\in[n_t]}$ realized from the induced data distribution $\mathcal{D}(\theta)$ and the target data distribution parameter ϕ as input to approximate the value $\mathsf{KL}(\phi||\varphi(\theta))$. We remark that such oracle has been widely used in the literature on KL divergence estimation (Rubenstein et al., 2019).

Definition 1 (Oracle EstimateKL)**.** *There exists an oracle* EstimateKL *that given any target parameter $\phi \in \Phi$, error tolerance $\epsilon_{\mathsf{KL}} > 0$ and error probability $p_{\mathsf{KL}} > 0$, and $N_{\mathsf{KL}}(\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})$ samples $z_1, \ldots, z_{N_{\mathsf{KL}}(\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})}$ from a distribution with parameter ϕ', returns an estimated* KL *divergence $\widetilde{\mathsf{KL}}(\phi||\phi')$ satisfying $\left|\widetilde{\mathsf{KL}}(\phi||\phi') - \mathsf{KL}(\phi||\phi')\right| \leq \epsilon_{\mathsf{KL}}$ with probability at least $1 - p_{\mathsf{KL}}$.*

With the oracle EstimateKL to approximately compute the KL divergence, we are now ready to present our inner algorithm LearnModel (see Algorithm 2).

Similar to before, for analysis purpose, we also define regret of LearnModel in S, the total number of rounds LearnModel has to go through in order to output a ϵ_{LM}-suboptimal model parameter w.r.t the KL objective function:

$$\mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})$$

Algorithm 2 Learn a model that approximately induces a given distribution parameter ϕ

function LearnModel($\phi \in \Phi$; $\epsilon_{\mathsf{LM}}, p_{\mathsf{LM}} > 0$, $\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}} > 0$)
  $S \leftarrow \frac{d_\Theta}{(\epsilon_{\mathsf{LM}} - \sqrt{\epsilon_{\mathsf{KL}} d_\Theta})^2}$, $\delta_{\mathsf{LM}} \leftarrow \sqrt{\epsilon_{\mathsf{KL}} d_\Theta}$
  $\eta_{\mathsf{LM}} \leftarrow \frac{1}{\sqrt{d_\Theta S}}$, $N_{\mathsf{KL}} \leftarrow N_{\mathsf{KL}}(\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})$
  $\theta_1 \leftarrow \mathbf{0}$
  for $s \leftarrow 1, \ldots, S$ **do**
    $u_s \leftarrow$ sample from $\mathrm{Unif}(\mathbb{S}^{d_\Theta})$
    $\theta_s^+ \leftarrow \theta_s + \delta_{\mathsf{LM}} u_s$, $\theta_s^- \leftarrow \theta_s - \delta_{\mathsf{LM}} u_s$
    $z_{s,1:N}^+ \sim \varphi(\theta_s^+)$, $z_{s,1:N}^- \sim \varphi(\theta_s^-)$ ▷ Deploy θ_s^+, θ_s^-; observe N_{KL} samples
    $\widetilde{\mathsf{KL}}\left(\phi||\varphi(\theta_s^+)\right) \leftarrow \mathsf{EstimateKL}(\phi, z_{s,1:N}^+, \epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})$
    $\widetilde{\mathsf{KL}}\left(\phi||\varphi(\theta_s^-)\right) \leftarrow \mathsf{EstimateKL}(\phi, z_{s,1:N}^-, \epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})$ ▷ Approximations of KL
    $\tilde{g}_s \leftarrow \frac{d_\Theta}{2\delta_{\mathsf{LM}}}\left(\widetilde{\mathsf{KL}}(\phi||\varphi(\theta_s^+)) - \widetilde{\mathsf{KL}}(\phi||\varphi(\theta_s^-))\right) \cdot u_s$ ▷ Approximation of $\nabla_\theta \mathsf{KL}(\phi||\varphi(\theta_s))$
    $\theta_{s+1} \leftarrow \Pi_{(1-\delta_{\mathsf{LM}})\Theta}(\theta_s - \eta_{\mathsf{LM}} \tilde{g}_s)$ ▷ Take gradient step and project
  $\bar{\theta} \leftarrow \frac{1}{S}\sum_{s=1}^S \theta_s$
  return $\bar{\theta}$

$$= \sum_{s=1}^S \left[\widetilde{\mathsf{KL}}(\phi||\varphi(\theta_s^+)) + \widetilde{\mathsf{KL}}(\phi||\varphi(\theta_s^-)) - 2\mathsf{KL}(\phi||\vartheta^*(\phi))\right]$$

where $\vartheta^*(\phi)$ is the model that can induce the target distribution ϕ. Using the similar arguments in Theorem 2, we first show the following regret guarantee for LearnModel:

Theorem 3 (High-probability regret bound for Algorithm 2 with S rounds)**.** *When* LearnModel *is run for S steps and invokes* EstimateKL *with arguments $\epsilon_{\mathsf{KL}} > 0$ and $p_{\mathsf{KL}} > 0$, we have $\forall p > 0$*

$$\mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL}) = O\left(\sqrt{d_\Phi S} + \sqrt{\epsilon_{\mathsf{KL}} d_\Phi} \cdot S + \sqrt{S \log \frac{1}{p}}\right)$$

with probability at least $1 - p - 2Sp_{\mathsf{KL}} > 0$.

Theorem 3 characterizes the regret as a function of the total number of deployments of the procedure in LearnModel. Together with regret characterization of the outer algorithm in Theorem 2, we can get the final regret bound.

4.3. Putting All Pieces Together

As shown in the previous section, both the outer algorithm (MinimizePR – in Section 4.1) and inner algorithm (LearnModel – in Section 4.2) achieve a sublinear regret w.r.t the total number of steps (T and S) when outputting an ϵ-optimal solutions. In this section, we combine the results in Section 4.1 and Section 4.2 to conclude the analysis for MinimizePR (Algorithm 1) for convex $\mathsf{PR}^\dagger(\phi)$. The main result of this section is summarized as follows:

Theorem 4 (Regret of MinimizePR in N)**.** *Under Assumption 2, and given access an oracle* EstimateKL*, there exists a choice of $\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}} > 0$ in Algorithm 2 such that for every*

$p > 0$,

$$\begin{aligned}&\mathcal{R}_N(\mathsf{MinimizePR}, \mathsf{PR})\\ =\ &\widetilde{O}\left((d_\Theta + d_\Phi) N_{\mathsf{KL}}(\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})^{1/6} N^{5/6} \sqrt{\log \frac{1}{p}}\right)\end{aligned}$$

with probability at least $1 - p$.

Proof Sketch of Theorem 4. Let T be the number of steps executed by the outer algorithm MinimizePR, and S the number of steps in LearnModel. Let $N_{\mathsf{KL}}(\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})$ (or N_{KL} for short) denote the number of samples used by EstimateKL. Since MinimizePR calls EstimatePR and LearnModel $2T$ times, and LearnModel calls EstimateKL $2S$ times, the overall number of samples involved in the whole process is $N = 2(2N_{\mathsf{KL}}S + 1)T$. Following the regret definition, we can break down the regret into the regret from calling EstimatePR in the outer algorithm and the regret from calling EstimateKL in LearnModel. Using the fact that $\mathsf{PR}^\dagger$ is Lipschitz in the distribution parameter ϕ and the distance between any two distribution parameters can be bounded by the KL divergence between the corresponding data distributions (Lemma 3), we show that the total regret in N can be expressed as:

$$\begin{aligned}&\mathcal{R}_N(\mathsf{MinimizePR}, \mathsf{PR})\\ =&O\left(\sqrt{N} + N_{\mathsf{KL}}T \cdot \sqrt{S \cdot \mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})}\right.\\ &\left.+(N_{\mathsf{KL}}S + 1) \cdot \mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR})\right)\end{aligned}$$

where $\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR})$ and $\mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})$ are obtained from Theorem 2 and Theorem 3 as functions of $\epsilon_{\mathsf{LM}}, \epsilon_{\mathsf{KL}}, S, T$ and D_Θ and D_Φ. Then by balancing the terms and setting ϵ_{LM} and ϵ_{KL} according to the convergence analysis for both MinimizePR and LearnModel (Claim 9 and Claim 10), we can get an express of the total regret. □

Theorem 4 show that our procedure is sublinear in the N, the total number of samples we deploy during the process. Notice that this also implies that our method is sublinear w.r.t. the total number of deployments $T_{\mathsf{total}} = S \times T$. To see this, recall that the total number of samples N required throughout the process is $N = 2(2N_{kl} \times S + 1)T$; if we measure the regret w.r.t. the total number of deployments, N_{kl} will be constant, and thus the regret will still be sublinear w.r.t. T_{total}. In addition, recall from Proposition 1, the lowest regret optimal classifier implies the optimal classifier up to some addictive error, which means that by having a sublinear regret in N (and T_{total}), we also get a model that is arbitrarily close to the performative optimal model.

Empirical Evaluation Using Toy Example We provide empirical results using a toy example to demonstrate the efficiency of our method. In particular, we compare our proposed method (which minimizes PR as a function of the distribution parameter ϕ after reparametrization) with the baseline method (which directly minimizes PR as a model parameter θ). We observe that under different settings, both methods converge. However, our proposed method (shown in orange) is more efficient: it demonstrates a much faster convergence rate on average over multiple runs, indicating that our reparametrization method is effective when dealing with distributions that are non-convex in θ but convex in ϕ (as per Assumption 1). The plot can be found in Appendix H. The details for reproducing our experimental results can be found at https://github.com/UCSC-REAL/PP-bandit-feedback.

5. Practical Consideration and Future Works

Since our method uses a double-loop zeroth order optimization method, the convergence is likely to be slow in practice. Thus, it requires extra consideration regarding the convergence rate, and efficiency can vary based on particular settings. In particular, the success of our method depends on the fast deployment of the frequently updated models. One potential way to speed up the deployment process may be performing *parallelization*, which accelerates the optimization process and reduces the time required to find a suitable solution (see, e.g., (Liu et al., 2020) for a detailed reference). In our algorithm, parallelization can be applied to several places, e.g., the two-point estimations can be computed in parallel and potentially speed up the process.

Acknowledgements

Y. Chen and Y. Liu are partially supported by the National Science Foundation (NSF) under grants IIS-2143895 and IIS-2040800.

Impact Statement

Since the process of finding the optimal performative model involves deploying sub-optimal models on human agents, sample efficiency is important. Thus sample complexity needs to be taken into consideration when choosing the particular KL divergence oracle used in LearnModel. Additionally, the concept of performativity highlights a significant broader impact of our work: the importance of recognizing scenarios where predictions can modify the very environment they're meant to predict. Take, for example, an online advertising platform utilizing machine learning models to tailor ads for users. These models, by analyzing user behaviors and traits to serve personalized ads, might inadvertently influence both user and advertiser actions, thereby establishing a feedback loop. This dynamic underscores the need for a mindful approach in deploying predictive models, especially in settings sensitive to the outcomes of such predictions.

References

Agarwal, A., Dekel, O., and Xiao, L. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Annual Conference on Learning Theory*, pp. 28–40, 2010.

Brown, G., Hod, S., and Kalemaj, I. Performative prediction in a stateful world. In *International Conference on Artificial Intelligence and Statistics*, pp. 6045–6061, 2022.

Chen, Y., Tang, Z., Zhang, K., and Liu, Y. Model transferability with responsive decision subjects. In *International Conference on Machine Learning*, pp. 4921–4952. PMLR, 2023.

Cutler, J., Drusvyatskiy, D., and Harchaoui, Z. Stochastic optimization under distributional drift, 2021.

Dong, J., Roth, A., Schutzman, Z., Waggoner, B., and Wu, Z. S. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 55–70, 2018.

Dong, R. and Ratliff, L. J. Approximate regions of attraction in learning with decision-dependent distributions, 2021.

Drusvyatskiy, D. and Xiao, L. Stochastic optimization with decision-dependent distributions. *arXiv preprint arXiv:2011.11173*, 2020.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *The Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 385–394, 2005.

Foster, D. J., Foster, D. P., Golowich, N., and Rakhlin, A. On the complexity of multi-agent decision making: From learning in games to partial monitoring. *arXiv preprint arXiv:2305.00684*, 2023.

Hardt, M., Megiddo, N., Papadimitriou, C., and Wootters, M. Strategic classification. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pp. 111–122, 2016.

Hardt, M., Jagadeesan, M., and Mendler-Dünner, C. Performative power. *arXiv preprint arXiv:2203.17232*, 2022.

Honorio, J. Lipschitz parametrization of probabilistic graphical models. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, 2011.

Izzo, Z., Ying, L., and Zou, J. How to learn when data reacts to your model: Performative gradient descent. In *International Conference on Machine Learning*, pp. 4641–4650, 2021.

Jagadeesan, M., Zrnic, T., and Mendler-Dünner, C. Regret minimization with performative feedback. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 2022.

Kleinberg, J. and Raghavan, M. How do classifiers induce agents to invest effort strategically? *ACM Trans. Econ. Comput.*, 8(4), oct 2020. ISSN 2167-8375.

Li, Q. and Wai, H.-T. State dependent performative prediction with stochastic approximation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3164–3186. PMLR, 2022.

Li, Q., Yau, C.-Y., and Wai, H.-T. Multi-agent performative prediction with greedy deployment and consensus seeking agents. *arXiv preprint arXiv:2209.03811*, 2022.

Liu, S., Chen, P.-Y., Kailkhura, B., Zhang, G., Hero, A., and Varshney, P. K. A primer on zeroth-order optimization in signal processing and machine learning, 2020.

Maheshwari, C., Chiu, C.-Y., Mazumdar, E., Sastry, S., and Ratliff, L. Zeroth-order methods for convex-concave min-max problems: Applications to decision-dependent risk minimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 6702–6734. PMLR, 2022.

Mendler-Dünner, C., Ding, F., and Wang, Y. Anticipating performativity by predicting from predictions. In *Advances in Neural Information Processing Systems*.

Mendler-Dünner, C., Perdomo, J., Zrnic, T., and Hardt, M. Stochastic optimization for performative prediction. *Advances in Neural Information Processing Systems*, 33, 2020.

Miller, J. P., Perdomo, J. C., and Zrnic, T. Outside the echo chamber: Optimizing the performative risk. In *International Conference on Machine Learning*, pp. 7710–7720, 2021.

Mofakhami, M., Mitliagkas, I., and Gidel, G. Performative prediction with neural networks. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research. PMLR, 25–27 Apr 2023.

Narang, A., Faulkner, E., Drusvyatskiy, D., Fazel, M., and Ratliff, L. J. Multiplayer performative prediction: Learning in decision-dependent games. *arXiv preprint arXiv:2201.03398*, 2022.

Perdomo, J., Zrnic, T., Mendler-Dünner, C., and Hardt, M. Performative prediction. In *International Conference on Machine Learning*, pp. 7599–7609, 2020.

Piliouras, G. and Yu, F.-Y. Multi-agent performative prediction: From global stability and optimality to chaos. *arXiv preprint arXiv:2201.10483*, 2022.

Ray, M., Ratliff, L. J., Drusvyatskiy, D., and Fazel, M. Decision-dependent risk minimization in geometrically decaying dynamic environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8081–8088, 2022.

Roth, A., Ullman, J., and Wu, Z. S. Watch and learn: Optimizing from revealed preferences feedback. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 949–962, 2016.

Rubenstein, P., Bousquet, O., Djolonga, J., Riquelme, C., and Tolstikhin, I. O. Practical and consistent estimation of f-divergences. In *Advances in Neural Information Processing Systems*, 2019.

Shamir, O. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

Wood, K. and Dall'Anese, E. Online saddle point tracking with decision-dependent data. *arXiv preprint arXiv:2212.02693*, 2022.

Yuan, M.-J. and Gao, W. Learning with interactive models over decision-dependent distributions. In *Proceedings of The 14th Asian Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 12–14 Dec 2023.

Zadimoghaddam, M. and Roth, A. Efficiently learning from revealed preference. In *International Workshop on Internet and Network Economics*, pp. 114–127. Springer, 2012.

A. Appendix Arrangement

We arrange the appendix as follows:

- Appendix B provides omitted algorithm and proofs for Section 2.
- Appendix C provides more examples that satisfies PR being convex in ϕ not in θ.
- Appendix D provides omitted example and proofs for Section 4.1.
- Appendix E provides omitted proofs for Section 4.2.
- Appendix F provides omitted proof for Section 4.3.
- Appendix G provides an additional literature review on performative prediction and related literature.
- Appendix H provides empirical verification using toy example generated by Example 3 to demonstrate the efficiency of our proposed method.

B. Omitted Algorithm and Proof for Section 2

B.1. Omitted Proof for Proposition 1

Proof. Let $x_1, \dots, x_n$ be the first n points queried by $\mathcal{A}$. By the convexity of f, the average of these points $\bar{x} = \frac{1}{n}\sum_i x_i$ satisfies

$$f(\bar{x}) - f(x^*) \le \frac{1}{n}\sum_{i=1}^{n}[f(x_i) - f(x^*)] = \frac{\mathcal{R}_n(\mathcal{A}, f)}{n}$$

Thus if $\mathcal{R}_n(\mathcal{A}, f) = o(n)$, then after $n = \mathcal{R}_n(\mathcal{A}, f)/\epsilon$ queries, $\bar{x}$ satisfies $f(\bar{x}) - f(x^*) \le \epsilon$ as required. □

B.2. Omitted algorithm and proof for Algorithm 3

Algorithm 3 is a straightforward generalization of the algorithm introduced by (Agarwal et al., 2010), while we generalize their setting where the function can be evaluated exactly to the setting where noisy evaluation is allowed.

Algorithm 3 Bandit algorithm for minimizing convex and lipschitz $\mathsf{PR}(\theta)$

function $\mathsf{EstimatePR}(\theta)$ ▷ Unbiased estimate of $\mathsf{PR}(\theta)$
  Deploy θ, observe sample $z \sim \mathcal{D}(\theta)$
  return $\ell(z;\theta)$
function MINIMIZEPR(T)
  $\delta \leftarrow \sqrt{d_\theta/T}$
  $\eta \leftarrow 1/\sqrt{d_\Theta T}$
  $\theta_1 \leftarrow \mathbf{0}$
  for $t \leftarrow 1, \dots, T$ **do**
    $u_t \leftarrow$ sample from $\mathrm{Unif}(\mathbb{S}^{d_\Theta})$
    $\theta_t^+ \leftarrow \theta_t + \delta u_t$, $\theta_t^- \leftarrow \theta_t - \delta u_t$
    $\widetilde{\mathsf{PR}}(\theta_t^+) \leftarrow \mathsf{EstimatePR}(\theta_t^+)$ ▷ Approximations of $\mathsf{PR}(\theta_t^+)$, $\mathsf{PR}(\theta_t^-)$
    $\widetilde{\mathsf{PR}}(\theta_t^-) \leftarrow \mathsf{EstimatePR}(\theta_t^-)$
    $g_t \leftarrow \frac{d_\Theta}{2\delta}\left(\widetilde{\mathsf{PR}}(\theta_t^+) - \widetilde{\mathsf{PR}}(\theta_t^-)\right) \cdot u_t$ ▷ Approximation of $\nabla_\theta \widehat{\mathsf{PR}}(\theta_t)$
    $\theta_{t+1} \leftarrow \Pi_{(1-\delta)\Theta}(\theta_t - \eta g_t)$ ▷ Take gradient step and project
  return $\frac{1}{T}\sum_{t=1}^{T}\theta_t$

To prove Lemma 1, we first provide a series of lemmas and claims that will be useful later.

Claim 1 (Regret from estimating PR)**.** *For any $p > 0$, with probability at least $1 - p$,*

$$\sum_{t=1}^{T}\left[\widetilde{\mathsf{PR}}(\theta_t^+) - f(\theta_t^+)\right] \le F\sqrt{T\log\frac{1}{p}} \quad \text{and} \quad \sum_{t=1}^{T}\left[\widetilde{\mathsf{PR}}(\theta_t^-) - f(\theta_t^-)\right] \le F\sqrt{T\log\frac{1}{p}}$$

Proof. The claim follows from Hoeffding's inequality, since $\mathsf{EstimatePR}$ is unbiased and bounded by $[0, F]$. □

Claim 2 (Regret from smoothing over the sphere or ball)**.** *For any $\theta \in \Theta$, $u \in \mathbb{S}$, and $\delta > 0$, all of the following are at most δL:*

$$|\mathsf{PR}(\theta + \delta u) - \mathsf{PR}(\theta)|, \quad |\mathsf{PR}(\theta - \delta u) - \mathsf{PR}(\theta)|,$$
$$\left|\frac{1}{2}[\mathsf{PR}(\theta + \delta u) + \mathsf{PR}(\theta - \delta u)] - \mathsf{PR}(\theta)\right|, \quad \textit{and} \quad |\widehat{\mathsf{PR}}(\theta) - \mathsf{PR}(\theta)|.$$

Proof sketch. Lipschitzness of PR. □

Claim 3 (Deviation of smoothed function)**.** *For any $p > 0$, with probability at least $1 - p$,*

$$\sum_{t=1}^{T}\widehat{\mathsf{PR}}(\theta_t) - \mathbb{E}_T\left[\sum_{t=1}^{T}\widehat{\mathsf{PR}}(\theta_t)\right] \le F\sqrt{T\log\frac{1}{p}}$$

Proof sketch. The left-hand side is the sum of a martingale difference sequence. The Azuma-Hoeffding inequality yields the result. □

Claim 4 (Gradient estimate is unbiased and bounded)**.** *There exists a constant $c > 0$ such that for all $t \in [T]$, $\mathbb{E}_t[g_t] = \nabla\widehat{\mathsf{PR}}(\theta_t)$ and $\|g_t\|_2^2 \le cd_\theta L^2$.*

Proof. Proved in Shamir (see Lemma 10, noting that the ℓ_2 norm is its own dual). □

Lemma 4 (Expected suboptimality under smoothing when PR is convex)**.** *Let $\theta \in \Theta$, and let $\theta_1, \dots, \theta_t \in \Theta$ be a sequence of iterates given by the update rule $\theta_{t+1} = \Pi_{(1-\delta)\theta}(\theta_t - \eta g_t) - \theta$ for some sequence of gradient estimates $g_t \in \mathbb{R}^{d_\Theta}$. Then*

$$\mathbb{E}_T\left[\sum_{t=1}^{T}\widehat{\mathsf{PR}}(\theta_t)\right] - \sum_{t=1}^{T}\widehat{\mathsf{PR}}(\theta) \le \frac{D_\Theta^2}{\eta} + \eta c d_\theta L^2 T$$

Proof of Lemma 4. Observe that

$$\begin{aligned}
\mathbb{E}_T\left[\sum_{t=1}^{T}\widehat{\mathsf{PR}}(\theta_t)\right] - \sum_{t=1}^{T}\widehat{\mathsf{PR}}(\theta) &= \sum_{t=1}^{T}\mathbb{E}_t\left[\widehat{\mathsf{PR}}(\theta_t) - \widehat{\mathsf{PR}}(\theta)\right] \\
&\le \sum_{t=1}^{T}\mathbb{E}_t\left[\nabla\widehat{\mathsf{PR}}(\theta_t)^\top(\theta_t - \theta)\right] && \text{(convexity of } \widehat{\mathsf{PR}}\text{)} \\
&= \sum_{t=1}^{T}\mathbb{E}_t\left[g_t^\top(\theta_t - \theta)\right] && \text{(Claim 4)}
\end{aligned}$$

To decompose $g_t^\top(\theta_t - \theta)$, note that

$$\begin{aligned}
\|\theta_{t+1} - \theta\|^2 &= \|\Pi_{(1-\delta)\theta}(\theta_t - \eta g_t) - x\|^2 \\
&\le \|\theta_t - \eta g_t - \theta\|^2 \\
&= \|\theta_t - \theta\|^2 + \eta^2\|g_t\|^2 - 2\eta \cdot g_t^\top(\theta_t - \theta)
\end{aligned}$$

Therefore

$$
\begin{aligned}
g_t^\top(\theta_t - x) &\le \frac{\|\theta_t - \theta\|^2 - \|\theta_{t+1} - \theta\|^2 + \eta^2\|g_t\|^2}{2\eta} \\
\sum_{t=1}^{T} \mathbb{E}_t\left[g_t^\top(\theta_t - \theta)\right] &\le \sum_{t=1}^{T} \mathbb{E}_t\left[\frac{\|\theta_t - \theta\|^2 - \|\theta_{t+1} - \theta\|^2 + \eta^2\|g_t\|^2}{2\eta}\right] \\
&\le \frac{1}{2\eta}\mathbb{E}_t\left[\|\theta_1 - \theta\|^2 + \eta^2 c d_\Theta L^2 T\right] && \text{(Claim 4)} \\
&\le \frac{D_\Theta^2}{2\eta} + \frac{\eta c d_\Theta L^2 T}{2} && \text{(diameter of } \Theta\text{)}
\end{aligned}
$$

as required. □

Claim 5 (Regret from projection)**.** *For any $\theta \in \Theta$, $\mathsf{PR}(\theta_\delta) - \mathsf{PR}(\theta) \le \delta D_\Theta L$.*

Proof. Since PR is L-Lipschitz and $\Pi_{(1-\delta)\Theta}$ projects from a set of diameter D_Θ to a set of diameter $(1-\delta)D_\Theta$, we have $\mathsf{PR}(\theta_\delta) - \mathsf{PR}(\theta) \le L\|\theta_\delta - \theta\| \le \delta D_\Theta L$. □

Claim 6 (Optimality of projected parameters)**.** *Since PR is convex in θ, $\mathsf{PR}\left(\Pi_{(1-\delta)\Theta}(\theta_{\texttt{OPT}})\right) = \operatorname{argmin}_{\theta \in (1-\delta)\Theta} \mathsf{PR}(\theta)$.*

Overall Regret Analysis for Lemma 1 We can now complete our regret bound for Lemma 1. Recall the lemma statement:

Lemma 1. *When $\mathsf{PR}(\theta)$ is convex, L-Lipschitz w.r.t. the deployed model parameter θ, there exists an algorithm (Algorithm 3) achieving $\mathcal{R}_N(\mathcal{A}_3, \mathsf{PR}) = O(\sqrt{d_\Theta N \log \frac{1}{p}})$ with probability at least $1-p$, where N is the total number of samples deployed during the process.*

Proof of Lemma 1. We have

$$
\begin{aligned}
\mathcal{R}_T(\mathcal{A}_3, f) &= \sum_{t=1}^{T}\left[\mathsf{EstimatePR}(\theta_t^+) + \mathsf{EstimatePR}(\theta_t^-) - 2\mathsf{PR}(\theta_{\texttt{OPT}})\right] \\
&= \underbrace{\sum_{t=1}^{T}\left[\widetilde{\mathsf{PR}}(\theta_t^+) + \widetilde{\mathsf{PR}}(\theta_t^-) - \mathsf{PR}(\theta_t^+) - \mathsf{PR}(\theta_t^-)\right]}_{\text{(I)}} + \underbrace{\sum_{t=1}^{T}\left[\mathsf{PR}(\theta_t^+) + \mathsf{PR}(\theta_t^-) - 2\widehat{\mathsf{PR}}(\theta_t)\right]}_{\text{(II)}} \\
&\quad + \underbrace{2\sum_{t=1}^{T}\left[\widehat{\mathsf{PR}}(\theta_t) - \mathbb{E}_t[\widehat{\mathsf{PR}}(\theta_t)]\right]}_{\text{(III)}} + \underbrace{2\sum_{t=1}^{T}\left[\mathbb{E}_t[\widehat{\mathsf{PR}}(\theta_t)] - \hat{f}(\theta_\delta^*)\right]}_{\text{(IV)}} \\
&\quad + \underbrace{2\sum_{t=1}^{T}\left[\widehat{\mathsf{PR}}(\theta_\delta^*) - f(\theta_\delta^*)\right]}_{\text{(V)}} + \underbrace{2\sum_{t=1}^{T}\left[\mathsf{PR}(\theta_\delta^*) - \mathsf{PR}(\theta_{\texttt{OPT}})\right]}_{\text{(VI)}} \\
&\le \underbrace{2F\sqrt{T\log\frac{1}{p_1}}}_{\substack{\text{(I), w.p. } 1-2p_1 \\ \text{(Claim 1)}}} + \underbrace{4\delta LT}_{\substack{\text{(II), w.p. 1} \\ \text{(Claim 2)}}} + \underbrace{2F\sqrt{T\log\frac{1}{p_2}}}_{\substack{\text{(III), w.p. } 1-2p_2 \\ \text{(Claim 3)}}} + \underbrace{\frac{2D_\Theta^2}{\eta} + 2\eta c d_\theta L^2 T}_{\substack{\text{(IV), w.p. 1} \\ \text{(Lemma 4)}}} + \underbrace{2\delta LT}_{\substack{\text{(V), w.p. 1} \\ \text{(Claim 2)}}} + \underbrace{2\delta D_\Theta LT}_{\substack{\text{(V), w.p. 1} \\ \text{(Claim 5)}}}
\end{aligned}
$$

Thus for any $p > 0$, a choice of $p_1 = p_2 = p/4$, along with $\eta = 1/\sqrt{d_\theta T}$ and any $\delta \le \sqrt{d_\theta/T}$, yields $\mathcal{R}_T(\mathcal{A}_3, \mathsf{PR}) = O(\sqrt{d_\theta T \log\frac{1}{p}})$ with probability at least $1-p$. Finally, since $\mathsf{EstimatePR}$ is queried twice per step, $n = 2T$, which gives us $\mathcal{R}_n(\mathcal{A}_3, \mathsf{PR}) = \mathcal{R}_T(\mathcal{A}_3, \mathsf{PR}) = O(\sqrt{d_\theta n \log\frac{1}{p}})$, completing the proof. □

C. Missing Proofs and Additional Examples for Section 3

Derivations for Example 3 Since ϕ is strictly increasing in $[0,1]$, the inverse mapping ϕ^{-1} is well-defined, and we can reformulate the performative risk $\mathsf{PR}(\theta)$ as a function of φ_θ, denoted $\mathsf{PR}^\dagger(\varphi_\theta)$, as follows:

$$\begin{aligned}\mathsf{PR}(\theta;x) &= \mathbb{E}_{y\sim\mathsf{Bern}(\varphi_\theta)}[\ell(\theta;x,y)]\\ &= \varphi_\theta\ell(\theta;x,1)+(1-\varphi_\theta)\ell(\theta;x,0)\\ &= \varphi_\theta\ell\left(\phi^{-1}(\varphi_\theta);x,1\right)+(1-\varphi_\theta)\ell\left(\phi^{-1}(\varphi_\theta);x,0\right)\\ &=: \mathsf{PR}^\dagger(\varphi_\theta;x)\end{aligned}$$

Plugging in ℓ, we have

$$\begin{aligned}\mathsf{PR}^\dagger(\varphi_\theta;x) &= -\varphi_\theta\cdot\left(\phi^{-1}(\varphi_\theta)x-1\right)^2-(1-\varphi_\theta)\cdot\left(\phi^{-1}(\varphi_\theta)x\right)^2\\ &= -\varphi_\theta\cdot\left(\sqrt{\varphi_\theta}x-1\right)^2-(1-\varphi_\theta)\varphi_\theta x^2 \qquad (\phi^{-1}(\varphi_\theta)=\sqrt{\varphi_\theta})\end{aligned}$$

Note that for all $x\in[0,1]$, $\mathsf{PR}^\dagger(\varphi_\theta;x)=\mathsf{PR}(\theta;x)$ is convex in φ_θ over $[0,1]$. In contrast,

$$\mathsf{PR}(\theta;x) = \theta^2\cdot\ell(\theta;x,1)+(1-\theta^2)\cdot\ell(\theta;x,0) \tag{6}$$

$$= -\theta^2\cdot(\theta x-1)^2-(1-\theta^2)\cdot(\theta x)^2 \tag{7}$$

which is non-convex in θ over $[0,1]$ for all $x\in[0,1]$.

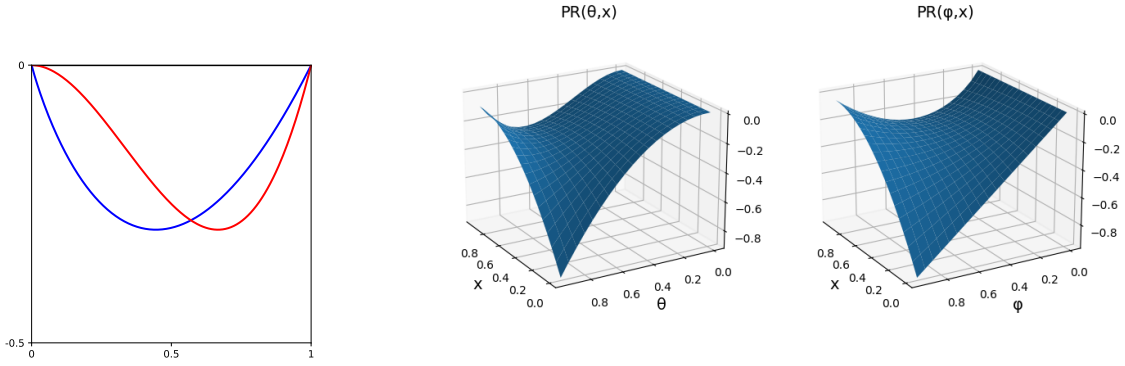

Figure 2. An example showing that our assumption is weaker than the mixture dominance assumption in Miller et al. (2021). In the left figure, the blue curve represents the function $\mathsf{PR}^\dagger(\varphi_\theta)$ which is convex w.r.t the data distribution parameter φ_θ; while the red curve represents the function $\mathsf{PR}(\theta)$, which is not a convex function with respect to θ. In the right two figures, we compare PR as a function of the model parameter θ and as a function of the distribution parameter ϕ.

Notice that Example 3 can be generalized to any distribution map $\phi(\theta)$ that satisfies $\phi(\theta)=\theta^\alpha$ for any $\alpha>1$, and any ℓ_β loss for even β value. Below in Figure 3, we provide the plot for for $\phi(\theta)=\theta^4$ with ℓ_4 norm loss (L_4 norm is defined as $L_4(x,y)=\left(\sum_{i=1}^d|x_i-y_i|^4\right)^{\frac{1}{4}}$ where d is the dimension of x and y). The original PR loss $\mathsf{PR}(\theta)$ is in red, which is non-convex), and the reformulated PR loss $\mathsf{PR}^\dagger(\theta)$ is in blue via reparameterization, which is convex).

In addition, Example 3 can also be a generalized monotone polynomial function $\phi(\theta)$. For example, $\phi(\theta)=c\theta^2+d\theta$, for $d\geq 0, d\geq -2c$.

Derivations for Example 4

$$\begin{aligned}\mathsf{PR}(\theta) &= \mathbb{E}_{x\sim\mathcal{D}(\theta)}\ell(x;\theta)\\ &= \mathbb{E}_{x\sim N(\varphi(\theta),\sigma^2)}(\theta x)^2\\ &= \theta^2(\sigma^2+\varphi(\theta)^2)\end{aligned}$$

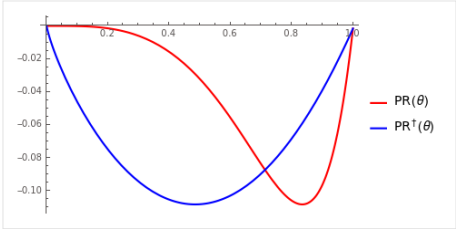

Figure 3. Another example showing PR is convex in ϕ but not θ. The original PR loss $\mathsf{PR}(\theta)$ is in red, which is non-convex), and the reformulated PR loss $\mathsf{PR}^\dagger(\theta)$ is in blue via reparameterization, which is convex)

when $\varphi(\theta) = \sqrt{\theta}$, $\mathsf{PR}(\theta)$ is not convex in θ. To see this:

$$\mathsf{PR}(\theta) = \theta^2(\sigma^2 + \theta)$$

On the other hand, since $\phi = \varphi(\theta) = \sqrt{\theta}$, we have

$$\mathsf{PR}(\phi) = \phi(\theta^2 + \phi)$$

which is convex in ϕ.

Derivations for Example 5

$$\begin{aligned}\mathsf{PR}(\theta) &= \mathbb{E}_{y\sim\mathcal{D}(\theta)}(\ell(\theta; x, y)) \\ &= -\mathbb{E}_{\text{Uniform}[0,\varphi(\theta)]}(\theta x - y)^2 \\ &= \theta^2 x^2 - \theta x\varphi(\theta) + \frac{1}{3}\varphi(\theta)^2\end{aligned}$$

when $\varphi(\theta) = \theta^2$, we have

$$\mathsf{PR}(\theta) = \theta^2 x^2 - \theta^3 x + \frac{1}{3}\theta^4$$

which is non-convex in θ. On the other hand, denote $\phi = \varphi(\theta)$, we have $\theta = \sqrt{\phi}$, plug it into PR, we have:

$$\mathsf{PR}(\phi) = \phi x^2 - \phi\sqrt{\phi} + \frac{1}{3}\phi^2$$

which is convex in ϕ.

D. Omitted Proof for Section 4.1

We present a series of lemmas and claims that are helpful for proving Theorem 2.

Claim 7 (Deviation of $\mathsf{PR}^\dagger$ due to error of LearnModel)**.** *If $\mathsf{PR}^\dagger$ is $L^\dagger$-Lipschitz, then for any $\phi \in \Phi$, the value $\hat{\theta} \in \Theta$ returned by* $\mathsf{LearnModel}(\phi, \epsilon_{\mathsf{LM}}, p_{\mathsf{LM}})$ *satisfies $|\mathsf{PR}^\dagger(\phi) - \mathsf{PR}(\hat{\theta})| \le L^\dagger\epsilon_{\mathsf{LM}}$ with probability at least $1 - p_{\mathsf{LM}}$.*

Proof. We have

$$\begin{aligned}\left|\mathsf{PR}^\dagger(\phi) - \mathsf{PR}(\hat{\theta})\right| &= \left|\mathsf{PR}^\dagger(\phi) - \mathsf{PR}^\dagger(\varphi(\hat{\theta}))\right| \\ &\le L^\dagger \left\|\phi - \varphi(\hat{\theta})\right\| && \text{(Lipschitzness of } \mathsf{PR}^\dagger\text{)} \\ &\le L^\dagger\epsilon_{\mathsf{LM}} && \text{(guarantee of LearnModel)}\end{aligned}$$

where the last inequality holds with probability at least $1 - p_{\mathsf{LM}}$. □

Claim 8 (Deviation of gradient estimate due to error of LearnModel and EstimatePR)**.** *Define*

$$\tilde{g}_t := \frac{d_\Phi}{\delta}\widetilde{\mathsf{PR}}(\hat{\theta}_t^+)u_t \qquad \text{and} \qquad g_t := \frac{d_\Phi}{\delta}\mathsf{PR}^\dagger(\phi_t^+)u_t \tag{8}$$

For any $t \in [T]$,

$$g_t - \tilde{g}_t \le \frac{d_\Phi}{\delta}\left[\mathsf{PR}(\hat{\theta}_t^+) - \widetilde{\mathsf{PR}}(\hat{\theta}_t^+) + \mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}(\hat{\theta}_t^+)\right]u_t.$$

Proof. We have

$$\begin{aligned}
g_t &= \frac{d_\Phi}{\delta}\mathsf{PR}^\dagger(\phi_t^+)u_t \\
&= \frac{d_\Phi}{\delta}\left[\widetilde{\mathsf{PR}}(\hat{\theta}_t^+) - \widetilde{\mathsf{PR}}(\hat{\theta}_t^+) + \mathsf{PR}(\hat{\theta}_t^+) - \mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}^\dagger(\phi_t^+)\right]u_t \\
&= \tilde{g}_t + \frac{d_\Phi}{\delta}\left[\mathsf{PR}(\hat{\theta}_t^+) - \widetilde{\mathsf{PR}}(\hat{\theta}_t^+) + \mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}(\hat{\theta}_t^+)\right]u_t && \text{(definition of } \tilde{g}_t\text{)}
\end{aligned}$$

□

Lemma 5 (Expected suboptimality under smoothing for $\mathsf{PR}^\dagger$)**.** *For any $\phi \in \Phi$, with probability at least $1 - Tp_{\mathsf{LM}}$ over the calls to* LearnModel*,*

$$\mathbb{E}_T\left[\sum_{t=1}^T \widehat{\mathsf{PR}}^\dagger(\phi_t)\right] - \sum_{t=1}^T \widehat{\mathsf{PR}}^\dagger(\phi) \le \frac{D_\Phi^2}{\eta} + \eta c d_\Phi L^2 T + \frac{D_\Phi L^\dagger \epsilon_{\mathsf{LM}} d_\Phi T}{\delta}$$

Proof of Lemma 5. For any $\phi \in \Phi$, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T \widehat{\mathsf{PR}}^\dagger(\phi_t)\right] - \sum_{t=1}^T \widehat{\mathsf{PR}}^\dagger(\phi) \\
&= \sum_{t=1}^T \mathbb{E}\left[\widehat{\mathsf{PR}}^\dagger(\phi_t) - \widehat{\mathsf{PR}}^\dagger(\phi)\right] \\
&\le \sum_{t=1}^T \mathbb{E}\left[\nabla\widehat{\mathsf{PR}}^\dagger(\phi_t)^\top(\phi_t - \phi)\right] && \text{(convexity of } \widehat{\mathsf{PR}}^\dagger\text{)} \\
&= \sum_{t=1}^T \mathbb{E}\left[g_t^\top(\phi_t - \phi)\right] && \text{(Claim 4)} \\
&= \sum_{t=1}^T \mathbb{E}\left[\left(\tilde{g}_t + \frac{d_Y}{\delta}\left[\mathsf{PR}(\hat{\theta}_t^+) - \widetilde{\mathsf{PR}}(\hat{\theta}_t^+) + \mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}(\hat{\theta}_t^+)\right]\cdot u_t\right)^\top(\phi_t - \phi)\right] && \text{(Claim 8)} \\
&= \sum_{t=1}^T \mathbb{E}\left[\left(\tilde{g}_t + \frac{d_Y}{\delta}\left[\mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}(\hat{\theta}_t^+)\right]\cdot u_t\right)^\top(\phi_t - \phi)\right] && (\mathbb{E}[\widetilde{\mathsf{PR}}(\cdot)] = \mathsf{PR}(\cdot) \text{ since EstimatePR is unbiased}) \\
&= \sum_{t=1}^T \mathbb{E}\left[\tilde{g}_t^\top(\phi_t - \phi)\right] + \frac{d_Y}{\delta}\sum_{t=1}^T \mathbb{E}\left[\left(\mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}(\hat{\theta}_t^+)\right)u_t^\top(\phi_t - \phi)\right] \\
&\le \sum_{t=1}^T \mathbb{E}\left[\tilde{g}_t^\top(\phi_t - \phi)\right] + \frac{d_Y}{\delta}\sum_{t=1}^T \mathbb{E}\left[\left|\mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}(\hat{\theta}_t^+)\right|\cdot\|u_t\|\cdot\|\phi_t - \phi\|\right] \\
&\le \sum_{t=1}^T \mathbb{E}\left[\tilde{g}_t^\top(\phi_t - \phi)\right] + \frac{d_Y}{\delta}\sum_{t=1}^T \mathbb{E}\left[L^\dagger\epsilon_h \cdot D_Y\right] && \text{(Claim 7, w.p. } 1 - Tp_h\text{)} \\
&\le \frac{D_Y^2}{\eta} + \eta c d_Y L^2 T + \frac{d_Y}{\delta}L^\dagger\epsilon_h D_Y T && \text{(same argument as in Lemma 4)}
\end{aligned}$$

□

Regret analysis for the outer algorithm in total number of step T We can now complete our regret bound for MinimizePR (Algorithm 1). We recall the theorem statement for Theorem 2:

Theorem 2 (High-probability regret bound for Algorithm 1 in T)**.** *When Algorithm 1 is called with arguments ϵ_{LM} and p_{LM}, we have for every $p > 0$ that*

$$\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR}) = O\left(\sqrt{d_\Phi T} + \sqrt{\epsilon_{\mathsf{LM}} d_\Phi} \cdot T + \sqrt{T \log \frac{1}{p}}\right)$$

with probability at least $1 - p - 2Tp_{\mathsf{LM}}$.

Proof of Theorem 2. We have

$$\begin{aligned}
&\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR}) \\
&= \sum_{t=1}^{T} \left[\mathsf{EstimatePR}(\hat{\theta}_t^+) + \mathsf{EstimatePR}(\hat{\theta}_t^-) - 2\mathsf{PR}(\theta_{\mathtt{OPT}})\right] \\
&= \underbrace{\sum_{t=1}^{T} \left[\widetilde{\mathsf{PR}}(\hat{\theta}_t^+) + \widetilde{\mathsf{PR}}(\hat{\theta}_t^-) - \mathsf{PR}(\hat{\theta}_t^+) - \mathsf{PR}(\hat{\theta}_t^-)\right]}_{\text{(I)}} + \underbrace{\sum_{t=1}^{T} \left[\mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}(\hat{\theta}_t^-) - \mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}^\dagger(\phi_t^-)\right]}_{\text{(II)}} \\
&\quad + \underbrace{\sum_{t=1}^{T} \left[\mathsf{PR}^\dagger(\phi_t^+) + \mathsf{PR}^\dagger(\phi_t^-) - 2\widehat{\mathsf{PR}}^\dagger(\phi_t)\right]}_{\text{(III)}} + \underbrace{2\sum_{t=1}^{T} \left[\widehat{\mathsf{PR}}^\dagger(\phi_t) - \mathbb{E}_t[\widehat{\mathsf{PR}}^\dagger(\phi_t)]\right]}_{\text{(IV)}} \\
&\quad + \underbrace{2\sum_{t=1}^{T} \left[\mathbb{E}_t[\widehat{\mathsf{PR}}^\dagger(\phi_t)] - \widehat{\mathsf{PR}}^\dagger(\phi_\delta^*)\right]}_{\text{(V)}} + \underbrace{2\sum_{t=1}^{T} \left[\widehat{\mathsf{PR}}^\dagger(\phi_\delta^*) - \mathsf{PR}^\dagger(\phi_\delta^*)\right]}_{\text{(VI)}} + \underbrace{2\sum_{t=1}^{T} \left[\mathsf{PR}^\dagger(\phi_\delta^*) - \mathsf{PR}^\dagger(\phi_{\mathtt{OPT}})\right]}_{\text{(VII)}} \\
&\leq \underbrace{2F\sqrt{T \log \frac{1}{p_1}}}_{\substack{\text{(I), w.p. } 1-2p_1 \\ \text{(Claim 1)}}} + \underbrace{2L^\dagger \epsilon_{\mathsf{LM}} T}_{\substack{\text{(II), w.p. } 1-2Tp_{\mathsf{LM}} \\ \text{(Claim 7)}}} + \underbrace{4\delta L T}_{\substack{\text{(III), w.p. } 1 \\ \text{(Claim 2)}}} + \underbrace{2F\sqrt{T \log \frac{1}{p_2}}}_{\substack{\text{(IV), w.p. } 1-2p_2 \\ \text{(Claim 3)}}} \\
&\quad + \underbrace{\frac{2D_\Phi^2}{\eta} + 2\eta c D_\Phi L^2 T + \frac{2D_\Phi L^\dagger \epsilon_{\mathsf{LM}} D_\Phi T}{\delta}}_{\substack{\text{(V), w.p. } 1-2Tp_{\mathsf{LM}} \\ \text{(Lemma 5)}}} + \underbrace{2\delta L^\dagger T}_{\substack{\text{(VI), w.p. } 1 \\ \text{(Claim 2)}}} + \underbrace{2\delta D_\Phi L^\dagger T}_{\substack{\text{(VII), w.p. } 1 \\ \text{(Claim 5)}}}
\end{aligned}$$

Recall that in Algorithm 1, we set $\delta = \sqrt{\epsilon_{\mathsf{LM}} D_\Phi}$ and $\eta = 1/\sqrt{D_\Phi T}$. Thus for any $p' > 0$, a choice of $p_1 = p_2 = p'/4$ yields

$$\mathcal{R}_T(\mathcal{A}_1, \mathsf{PR}) = O\left(\sqrt{D_\Phi T} + \sqrt{\epsilon_{\mathsf{LM}} D_\Phi} \cdot T + \sqrt{T \log \frac{1}{p'}}\right)$$

with probability at least $1 - p' - 2Tp_{\mathsf{LM}}$ as required. □

E. Omitted Proof for Section 4.2

We first provide a proof for Lemma 2. Recall the lemma statement:

Lemma 2 (Lipschitzness of $\mathsf{KL}(\phi||\varphi(\theta))$ in θ)**.** *Given two (ℓ_2, K)-Lipschitz continuous distributions $\mathcal{D}_1 = p(\cdot \mid \varphi(\theta_1))$ and $\mathcal{D}_2 = p(\cdot \mid \varphi(\theta_2))$, and a target distribution parameter $\phi \in \Phi$, we have $|\mathsf{KL}(\phi||\varphi(\theta_1)) - \mathsf{KL}(\phi||\varphi(\theta_2))| \leq L_{\mathsf{KL}} \|\theta_1 - \theta_2\|$ with a constant $L_{\mathsf{KL}} > 0$.*

Proof of Lemma 2.

$$
\begin{aligned}
&|\mathsf{KL}(\phi||\varphi(\theta_1)) - \mathsf{KL}(\phi||\varphi(\theta_2))| \\
=&\left|\int_z p(z|\phi)\log\frac{p(z|\phi)}{p(z|\varphi(\theta_1))}dz - \int_z p(z|\phi)\log\frac{p(z|\phi)}{p(z|\varphi(\theta_2))}dz\right| \\
=&\left|\int_z p(z|\phi)(\log p(z|\varphi(\theta_1)) - \log p(z|\varphi(\theta_2)))dz\right| \\
\le&\int_z p(z|\phi)\left|\log p(z|\varphi(\theta_1)) - \log p(z|\varphi(\theta_2))\right|dz \\
\le&\int_z p(z|\phi)L_{\mathsf{KL}}\|\theta_1-\theta_2\|dz \qquad (\mathcal{P}_1 \text{ and } \mathcal{P}_2 \text{ are lipschitzness continuous, Theorem 3 of (Honorio, 2011))} \\
=&L_{\mathsf{KL}}\|\theta_1-\theta_2\|\underbrace{\int_z p(z|\phi)dz}_{=1} \\
=&L_{\mathsf{KL}}\|\theta_1-\theta_2\|
\end{aligned}
$$

□

Next, we provide the proof for Lemma 3. Recall the lemma statement:

Lemma 3. *With Assumption 2c, we have* $\|\phi_1-\phi_2\| \le L_\phi\sqrt{\mathsf{KL}(\phi_1||\phi_2)}$ *for some constant* $L_\phi > 0$.

Proof of Lemma 3.

$$
\|\phi_1-\phi_2\|_2 \le L_{\mathsf{TV}}d_{\mathsf{TV}}(\phi_1,\phi_2) \le L_{\mathsf{TV}}\sqrt{\frac{1}{2}\mathsf{KL}(\phi_1,\phi_2)} \triangleq L_\phi\sqrt{\mathsf{KL}(\phi_1,\phi_2)}
$$

The second inequality is due to Pinsker's inequality. □

We then show the example provide by Example 1 is convex in θ. Recall the example:

Example 1. *Consider the density function* $p(z;\varphi(\theta))$ *of the data distribution* $\mathcal{D}(\theta)$ *satisfying* $p(z;\varphi(\theta)) = \mathrm{Unif}(\exp(c\varphi(\theta)))$ *for some constant* $c > 0$ *and for any convex function* $\varphi(\theta)$*, then* $\mathsf{KL}(\phi||\varphi(\cdot))$ *is convex over* θ.

Below we provide proof for it being convex in θ:

Proof for Example 1 being convex in θ. Under condition 1, we have $p(z;\phi) = \frac{1}{\exp(c\varphi(\theta))}$. We can rewrite the $\mathsf{KL}(\phi||\varphi(\theta))$ divergence as:

$$
\begin{aligned}
\mathsf{KL}(\phi||\varphi(\theta)) &= \int_z p(z;\phi)\log\frac{p(z;\phi)}{p(z;\varphi(\theta))}dz \\
&= \int_z \frac{1}{\exp(c\phi)}\log\frac{\exp(c\varphi(\theta))}{\exp(c\phi)}dz \\
&= \frac{\exp(c\varphi(\theta))}{\exp(c\phi)}\log\frac{\exp(c\varphi(\theta)}{\exp(c\phi)} \\
&= \exp(c(\varphi(\theta)-\phi))c(\varphi(\theta)-\phi)
\end{aligned}
$$

Denote $\mathsf{KL}(\phi||\varphi(\theta)) = f(g(\theta))$ where $f(x) = cx\exp(cx)$ and $g(\theta) = \varphi(\theta) - \phi$.

To show Equation (5) is convex in θ, it suffices to show f(x) is convex non-decreasing in x, and $g(\theta)$ is convex in θ. First, $g(\theta)$ is convex in θ due to condition 2.

For $f(x)$, take the first and second derivative and find conditions to make them both non negative:

$$\begin{aligned}\frac{\partial f(x)}{\partial x} &= c\exp(cx) + cx^2\exp(cx)\\ &= c\exp(cx)(1+cx) \geq 0\\ \frac{\partial^2 f(x)}{\partial x^2} &= c^2\exp(cx)(2+cx) \geq 0\end{aligned}$$

It suffices to set $(2+cx) \geq 0$ and $c(1+cx) \geq 0$ which suffices to set $c \geq \frac{2}{\max|\varphi(\theta)-\phi|}$.

□

Regret Analysis and convergence guarantee of LearnModel **in total number of steps** S We can now complete our regret bound for LearnModel (Algorithm 2). Recall the theorem statement:

Theorem 3 (High-probability regret bound for Algorithm 2 with S rounds)**.** *When* LearnModel *is run for S steps and invokes* EstimateKL *with arguments $\epsilon_{\mathsf{KL}} > 0$ and $p_{\mathsf{KL}} > 0$, we have $\forall p > 0$*

$$\mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL}) = O\left(\sqrt{d_\Phi S} + \sqrt{\epsilon_{\mathsf{KL}} d_\Phi}\cdot S + \sqrt{S\log\frac{1}{p}}\right)$$

with probability at least $1 - p - 2Sp_{\mathsf{KL}} > 0$.

Proof of Theorem 3.

$$\begin{aligned}&\mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})\\ &= \sum_{s=1}^{S}\left[\widetilde{\mathsf{KL}}(\phi||\varphi(\theta_s^+)) + \widetilde{\mathsf{KL}}(\phi||\varphi(\theta_s^-)) - 2\underbrace{\mathsf{KL}(\phi||\varphi(\vartheta^*(\phi)))}_{=0,\varphi(\vartheta^*(\phi)))=\phi}\right]\\ &= \underbrace{\sum_{s=1}^{S}\left[\widetilde{\mathsf{KL}}(\phi||\varphi(\theta_s^+)) - \mathsf{KL}(\phi||\varphi(\theta_s^+)) + \widetilde{\mathsf{KL}}(\phi||\varphi(\theta_s^+)) - \mathsf{KL}(\phi||\varphi(\theta_s^-))\right]}_{\text{(I)}}\\ &\quad + \underbrace{\sum_{s=1}^{S}\left[\mathsf{KL}(\phi||\varphi(\theta_s^+)) + \mathsf{KL}(\phi||\varphi(\theta_s^-)) - 2\widehat{\mathsf{KL}}(\phi||\varphi(\theta_s))\right]}_{\text{(II)}}\\ &\quad + \underbrace{2\sum_{s=1}^{S}\left[\widehat{\mathsf{KL}}(\phi||\varphi(\theta_s)) - \mathbb{E}_s[\widehat{\mathsf{KL}}(\phi||\varphi(\theta_s))]\right]}_{\text{(III)}} + \underbrace{2\sum_{s=1}^{S}\left[\mathbb{E}_s[\widehat{\mathsf{KL}}(\phi||\varphi(\theta_s))] - \widehat{\mathsf{KL}}(\phi||\varphi(\theta_\delta^*))\right]}_{\text{(IV)}}\\ &\quad + \underbrace{2\sum_{s=1}^{S}\left[\widehat{\mathsf{KL}}(\phi||\varphi(\theta_\delta^*)) - \mathsf{KL}(\phi||\varphi(\theta_\delta^*))\right]}_{\text{(V)}} + \underbrace{2\sum_{s=1}^{S}\left[\mathsf{KL}(\phi||\varphi(\theta_\delta^*)) - \mathsf{KL}(\phi||\varphi(\theta^*))\right]}_{\text{(VI)}}\\ &\leq \underbrace{2\epsilon_{\mathsf{KL}}S}_{\substack{\text{(I), w.p. } 1-2Sp_{\mathsf{KL}}\\ \text{(Assumption 1)}}} + \underbrace{4\delta L_{\mathsf{KL}}S}_{\substack{\text{(II), w.p. 1}\\ \text{(Claim 2)}}} + \underbrace{2F_{\mathsf{KL}}\sqrt{S\log\frac{1}{p_2}}}_{\substack{\text{(III), w.p. } 1-2p_2\\ \text{(Claim 3)}}}\\ &\quad + \underbrace{\frac{2D_\Theta^2}{\eta_{\mathsf{LM}}} + 2\eta_{\mathsf{LM}}d_\Theta L_{\mathsf{KL}}^2 S + \frac{2D_\Theta L_{\mathsf{KL}}\epsilon_{\mathsf{KL}}d_\Theta S}{\delta_{\mathsf{LM}}}}_{\substack{\text{(IV), w.p. } 1-2Sp_{\mathsf{KL}}\\ \text{(Similar argument as Lemma 5)}}} + \underbrace{2\delta_{\mathsf{LM}}L_{\mathsf{KL}}S}_{\substack{\text{(V), w.p. 1}\\ \text{(Claim 2)}}} + \underbrace{2\delta_{\mathsf{LM}}D_\Theta L_{\mathsf{KL}}S}_{\substack{\text{(VI), w.p. 1}\\ \text{(Claim 5)}}}\end{aligned}$$

Similar to Algorithm 1, we set $\delta_{\mathsf{LM}} = \sqrt{\epsilon_{\mathsf{KL}} d_\Theta}$, $\eta_{\mathsf{LM}} = 1/\sqrt{d_\Theta S}$. For any $p_2 = p'/2 > 0$, it yields

$$R_S(\mathsf{LearnModel}, \mathsf{KL}) = O\left(\sqrt{d_\Theta S} + \sqrt{\epsilon_{\mathsf{KL}} d_\theta} S + \sqrt{S \log \frac{1}{p}}\right)$$

with probability $1 - p' - 2Sp_{\mathsf{KL}}$ ¿ 0.

□

F. Omitted Proof for Section 4.3

We start with leveraging Theorem 2 to show the following convergence guarantee for MinimizePR (Algorithm 1).

Claim 9 (Convergence of MinimizePR)**.** *Given any $\epsilon, p > 0$,* MinimizePR *outputs an ϵ-suboptimal solution for* $\mathsf{PR}(\theta)$ *with probability at least $1 - p$. Moreover,* MinimizePR *runs for $T = O(d_\Phi/\epsilon^2)$ steps and performs $O(d_\Phi/\epsilon^2)$ queries to* EstimatePR*, as well as $O(d_\Phi/\epsilon^2)$ queries to* LearnModel *with $\epsilon_{\mathsf{LM}} = O(\epsilon^2)$ and $p_{\mathsf{LM}} = O(\epsilon^2 p/d_\Phi)$.*

Proof of Claim 9. Choosing $\epsilon_{\mathsf{LM}} = 1/T$, $p_{\mathsf{LM}} = p/2T$, and $p' = p/2$, Theorem 2 shows that MinimizePR satisfies

$$\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR}) = O\left(\sqrt{d_\Phi T}\right)$$

with probability $1 - p$, using $2T$ queries to EstimatePR and $2T$ queries to LearnModel. By Proposition 1, $T = O(d_\Phi/\epsilon^2)$ steps suffice to output a model that is ϵ-suboptimal with respect to PR. Plugging in this bound on T into the expressions for ϵ_{LM} and p_{LM} above yields the result. □

Similarly, we have the convergence guarantee for LearnModel as well:

Claim 10 (Convergence of LearnModel)**.** *Given any $\phi \in \Phi$ and $\epsilon_{\mathsf{LM}}, p_{\mathsf{LM}} > 0$,* LearnModel *outputs an ϵ_{LM}-suboptimal model for Equation (5) with probability at least $1 - p_{\mathsf{LM}}$. Moreover,* LearnModel *runs for $S = O(d_\Theta/\epsilon_{\mathsf{LM}}^2)$ steps and performs two queries to* EstimateKL *per step with $N_{\mathsf{KL}}(\frac{\epsilon_{\mathsf{LM}}^2}{d_\theta}, \frac{\epsilon_{\mathsf{LM}}^2 p_{\mathsf{LM}}}{4 d_\theta})$ samples per query.*

Proof of Claim 10. Choosing $\epsilon_{\mathsf{KL}} = 1/S$, $p_{\mathsf{KL}} = p_{\mathsf{LM}}/4S$ and $p' = p_{\mathsf{LM}}/2$, Theorem 3 shows that LearnModel satisfies

$$\mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL}) = O\left(\sqrt{d_\Phi S}\right)$$

By Proposition 1, $S = O(d_\Theta/\epsilon_{\mathsf{LM}}^2)$ steps suffice to output a model that is ϵ_{LM}-suboptimal with respect to KL; thus we have $\epsilon_{\mathsf{KL}} = \frac{\epsilon_{\mathsf{LM}}^2}{d_\Theta}$, $p_{\mathsf{KL}} = \frac{1}{4S p_{\mathsf{LM}}}$. In total, LearnModel makes $2S$ queries to EstimateKL with $N_{\mathsf{KL}}(\frac{\epsilon_{\mathsf{LM}}^2}{d_\Theta}, \frac{\epsilon_{\mathsf{LM}}^2 p_{\mathsf{LM}}}{4 d_\theta})$ samples per query. □

Now are are ready to prove Theorem 4. Recall the theorem statement:

Theorem 4 (Regret of MinimizePR in N)**.** *Under Assumption 2, and given access an oracle* EstimateKL*, there exists a choice of $\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}} > 0$ in Algorithm 2 such that for every $p > 0$,*

$$\begin{aligned} &\mathcal{R}_N(\mathsf{MinimizePR}, \mathsf{PR}) \\ &= \widetilde{O}\left((d_\Theta + d_\Phi) N_{\mathsf{KL}}(\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})^{1/6} N^{5/6} \sqrt{\log \frac{1}{p}}\right) \end{aligned}$$

with probability at least $1 - p$.

Proof of Theorem 4. Let T be the number of steps executed by MinimizePR, and S the number of steps in LearnModel. Let $N_{\mathsf{KL}}(\epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})$ (or N_{KL} for short) denote the number of samples used by $\mathsf{EstimateKL}(\cdot, \cdots, \epsilon_{\mathsf{KL}}, p_{\mathsf{KL}})$. Since MinimizePR calls EstimatePR and LearnModel $2T$ times, and LearnModel calls EstimateKL $2S$ times, the overall number of samples is $N = 2(2N_{\mathsf{KL}} S + 1)T$.

Let $\theta_{t,s}^+, \theta_{t,s}^-$ denote the models deployed by EstimateKL in the s-th step of LearnModel within the t-th step of MinimizePR, obtaining samples $z_{t,s,1}^+, \ldots, z_{t,s,N_{\mathsf{KL}}}^+$ and $z_{t,s,1}^-, \ldots, z_{t,s,N_{\mathsf{KL}}}^-$, respectively. Similarly, let $\hat{\theta}_t^+, \hat{\theta}_t^-$ denote the models deployed by EstimatePR in the t-th step of MinimizePR, obtaining samples $\hat{z}_t^+, \hat{z}_t^-$.

The total regret can be written as

$$
\begin{aligned}
&\mathcal{R}_N(\mathsf{MinimizePR}, \mathsf{PR}) \\
&= \sum_{t=1}^{T}\left(\ell(\hat{z}_t^+;\hat{\theta}_t^+) + \ell(\hat{z}_t^-;\hat{\theta}_t^-) - 2\mathsf{PR}(\theta^*) + \sum_{s=1}^{S}\sum_{i=1}^{N_{\mathsf{KL}}}\left[\ell(z_{t,s,i}^+;\theta_{t,s}^+) + \ell(z_{t,s,i}^-;\theta_{t,s}^-) - 2\mathsf{PR}(\theta^*)\right]\right) \\
&= \underbrace{\sum_{t=1}^{T}\left(\ell(\hat{z}_t^+;\hat{\theta}_t^+) - \mathsf{PR}(\hat{\theta}_t^+) + \ell(\hat{z}_t^-;\hat{\theta}_t^-) - \mathsf{PR}(\hat{\theta}_t^-) + \sum_{s=1}^{S}\sum_{i=1}^{N_{\mathsf{KL}}}\left[\ell(z_{t,s,i}^+;\theta_{t,s}^+) - \mathsf{PR}(\theta_{t,s}^+) + \ell(z_{t,s,i}^-;\theta_{t,s}^-) - \mathsf{PR}(\theta_{t,s}^-)\right]\right)}_{n \text{ difference terms with expectation zero}} \\
&\qquad + \sum_{t=1}^{T}\left(\mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}(\hat{\theta}_t^-) - 2\mathsf{PR}(\theta^*) + \sum_{s=1}^{S}\sum_{i=1}^{N_{\mathsf{KL}}}\left[\mathsf{PR}(\theta_{t,s}^+) + \mathsf{PR}(\theta_{t,s}^-) - 2\mathsf{PR}(\theta^*)\right]\right) \\
&= O\left(\sqrt{N}\right) + \sum_{t=1}^{T}\left(\mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}(\hat{\theta}_t^-) - 2\mathsf{PR}(\theta^*) + \sum_{s=1}^{S}\sum_{i=1}^{N_{\mathsf{KL}}}\left[\mathsf{PR}(\theta_{t,s}^+) + \mathsf{PR}(\theta_{t,s}^-) - 2\mathsf{PR}(\theta^*)\right]\right) \\
&\qquad\qquad \text{(by Hoeffding's inequality, w.p. } 1-p') \\
&= O\left(\sqrt{N}\right) + \sum_{t=1}^{T}\left[\mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}(\hat{\theta}_t^-) - 2\mathsf{PR}(\theta^*)\right] \\
&\qquad + N_{\mathsf{KL}}\cdot\sum_{t=1}^{T}\sum_{s=1}^{S}\left[(\mathsf{PR}(\theta_{t,s}^+) + \mathsf{PR}(\theta_{t,s}^-)) - (\mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}(\hat{\theta}_t^-)) + (\mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}(\hat{\theta}_t^-)) - 2\mathsf{PR}(\theta^*)\right] \\
&= O\left(\sqrt{N}\right) + (N_{\mathsf{KL}}S+1)\sum_{t=1}^{T}\left[\mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}(\hat{\theta}_t^-) - 2\mathsf{PR}(\theta^*)\right] \\
&\qquad + N_{\mathsf{KL}}\cdot\sum_{t=1}^{T}\sum_{s=1}^{S}\left[\mathsf{PR}(\theta_{t,s}^+) - \mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}(\theta_{t,s}^-) - \mathsf{PR}(\hat{\theta}_t^-)\right] \\
&= O\left(\sqrt{N}\right) + (N_{\mathsf{KL}}S+1)\cdot\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR}) + N_{\mathsf{KL}}\cdot\sum_{t=1}^{T}\sum_{s=1}^{S}\left[\mathsf{PR}(\theta_{t,s}^+) - \mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}(\theta_{t,s}^-) - \mathsf{PR}(\hat{\theta}_t^-)\right] \\
&= O\left(\sqrt{N}\right) + (N_{\mathsf{KL}}S+1)\cdot\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR}) + N_{\mathsf{KL}}\cdot\sum_{t=1}^{T}\sum_{s=1}^{S}\left[\mathsf{PR}^\dagger(\varphi(\theta_{t,s}^+)) - \mathsf{PR}(\hat{\theta}_t^+) + \mathsf{PR}^\dagger(\varphi(\theta_{t,s}^-)) - \mathsf{PR}(\hat{\theta}_t^-)\right] \\
&= O\left(\sqrt{N}\right) + (N_{\mathsf{KL}}S+1)\cdot\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR}) \\
&\qquad + N_{\mathsf{KL}}\cdot\underbrace{\sum_{t=1}^{T}\sum_{s=1}^{S}\left[\mathsf{PR}^\dagger(\varphi(\theta_{t,s}^+)) - \mathsf{PR}^\dagger(\phi_t^+)\right]}_{\text{(I)}} + N_{\mathsf{KL}}\cdot\underbrace{\sum_{t=1}^{T}\sum_{s=1}^{S}\left[\mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}(\hat{\theta}_t^+)\right]}_{\text{(II)}} \\
&\qquad + N_{\mathsf{KL}}\cdot\underbrace{\sum_{t=1}^{T}\sum_{s=1}^{S}\left[\mathsf{PR}^\dagger(\varphi(\theta_{t,s}^-)) - \mathsf{PR}^\dagger(\phi_t^-)\right]}_{\text{(III)}} + N_{\mathsf{KL}}\cdot\underbrace{\sum_{t=1}^{T}\sum_{s=1}^{S}\left[\mathsf{PR}^\dagger(\phi_t^-) - \mathsf{PR}(\hat{\theta}_t^-)\right]}_{\text{(IV)}}
\end{aligned}
$$

Term (I) is:

$$
\begin{aligned}
\sum_{t=1}^{T}\sum_{s=1}^{S}\left[\mathsf{PR}^\dagger(\varphi(\theta_{t,s}^+)) - \mathsf{PR}^\dagger(\phi_t^+)\right] &\le L^\dagger\cdot\sum_{t=1}^{T}\sum_{s=1}^{S}\left\|\varphi(\theta_{t,s}^+) - \phi_t^+\right\| && \text{(Lipschitzness of } \mathsf{PR}^\dagger) \\
&\le L^\dagger\cdot\sum_{t=1}^{T}\sqrt{S\sum_{s=1}^{S}\left(\left\|\varphi(\theta_{t,s}^+) - \phi_t^+\right\|^2\right)} && \text{(Cauchy-Schwarz)}
\end{aligned}
$$

$$= L^\dagger T \sqrt{S \sum_{s=1}^{S} L_\theta^2 \mathsf{KL}(\phi_t^+ || \varphi(\theta_{t,s}^+))} \qquad \text{(Lemma 2)}$$
$$\leq L^\dagger L_\theta T \cdot \sqrt{S \cdot \mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})}$$

and term (III) is analogous. Term (II) is

$$\sum_{t=1}^{T}\sum_{s=1}^{S}\left[\mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}(\hat{\theta}_t^+)\right] = S \cdot \sum_{t=1}^{T}\left[\mathsf{PR}^\dagger(\phi_t^+) - \mathsf{PR}^\dagger(\varphi(\hat{\theta}_t^+))\right]$$
$$\leq L^\dagger S \cdot \sum_{t=1}^{T}\left\|\phi_t^+ - \varphi(\hat{\theta}_t^+)\right\| \qquad \text{(Lipschitzness of } \mathsf{PR}^\dagger\text{)}$$
$$\leq L^\dagger S \cdot \sum_{t=1}^{T} L_\theta \sqrt{\mathsf{KL}(\phi_t^+ || \varphi(\hat{\theta}_t^+))} \qquad \text{(Lemma 2)}$$
$$\leq L^\dagger L_\theta \cdot S \cdot \sum_{t=1}^{T} \sqrt{\frac{1}{S}\sum_{s=1}^{S} \mathsf{KL}(\phi_t^+ || \varphi(\theta_{t,s}^+))}$$
$$(\hat{\theta}_t^+ := \tfrac{1}{S}\textstyle\sum_{s=1}^{S}\theta_{t,s}^+, \text{ convexity of } \mathsf{KL}(\phi_t^+||\varphi(\theta)))$$
$$\leq L^\dagger L_\phi T S \sqrt{\frac{1}{S}\mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})}$$
$$= L^\dagger L_\phi T \sqrt{S \cdot \mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})}$$

and term (IV) is analogous. In total we have

$$\mathcal{R}_N(\mathsf{MinimizePR}, \mathsf{PR})$$
$$= O\left(\sqrt{N} + N_{\mathsf{KL}} T \cdot \sqrt{S \cdot \mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})} + (N_{\mathsf{KL}} S + 1) \cdot \mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR})\right)$$
$$= N \cdot O\left(\frac{1}{\sqrt{N}} + \sqrt{\frac{\mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})}{S}} + \frac{\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR})}{T}\right) \qquad (n = 2(N_{\mathsf{KL}} 2S + 1)T)$$
$$= N \cdot O\left(\frac{1}{\sqrt{N}} + \sqrt{\sqrt{\frac{d_\Theta \log \frac{1}{p'}}{S}} + \sqrt{\epsilon_{\mathsf{KL}} d_\Theta}} + \sqrt{\frac{d_\Phi \log \frac{1}{p''}}{T}} + \sqrt{\epsilon_{\mathsf{LM}} d_\Phi}\right)$$
(by Theorem 2,Theorem 3, w.p. to be analyzed later)
$$= N \cdot O\left(\left(\frac{d_\Theta}{S}\log\frac{1}{p'}\right)^{1/4} + (\epsilon_{\mathsf{KL}} d_\Theta)^{1/4} + \left(\frac{d_\Phi}{T}\log\frac{1}{p''}\right)^{1/2} + (\epsilon_{\mathsf{LM}} d_\Phi)^{1/2}\right)$$
(for $a, b \geq 0$, $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$; $\frac{1}{\sqrt{n}} \leq \sqrt{\frac{d_\Phi}{T}}$)
$$\leq N \cdot \left(1 + \left(\log\frac{1}{p'}\right)^{1/4} + \left(\log\frac{1}{p''}\right)^{1/2}\right) \cdot O\left(\left(\frac{d_\Theta}{S}\right)^{1/4} + (\epsilon_{\mathsf{KL}} d_\Theta)^{1/4} + \left(\frac{d_\Phi}{T}\right)^{1/2} + (\epsilon_{\mathsf{LM}} d_\Phi)^{1/2}\right)$$
$$= N \cdot \left(1 + \left(\log\frac{1}{p'}\right)^{1/4} + \left(\log\frac{1}{p''}\right)^{1/2}\right) \cdot O\left(\left(\frac{d_\Theta}{S}\right)^{1/4} + (\epsilon_{\mathsf{KL}} d_\Theta)^{1/4} + \left(\frac{d_\Phi N_{\mathsf{KL}} S}{N}\right)^{1/2} + (\epsilon_{\mathsf{LM}} d_\Phi)^{1/2}\right)$$
$(T = \frac{N}{N_{\mathsf{KL}} S + 1})$

Choose $\epsilon_{\mathsf{LM}} = \left(\frac{N_{\mathsf{KL}}}{N}\right)^{1/3}$ and $\epsilon_{\mathsf{KL}} = \frac{1}{4d_\Theta}\left(\frac{N_{\mathsf{KL}}}{N}\right)^{2/3}$.

To balance the terms, set the number of steps for the outer algorithm to be $T = \frac{d_\Phi}{(\epsilon - \sqrt{\epsilon_{\mathsf{LM}} d_\Phi})^2}$, and the number of steps in

LearnModel to be

$$S = \frac{d_\Theta}{\left(\epsilon_{\mathsf{LM}} - \sqrt{\epsilon_{\mathsf{KL}} d_\Theta}\right)^2} = 4 d_\Theta \left(\frac{N}{N_{\mathsf{KL}}}\right)^{2/3}$$

Plugging these expressions for ϵ_{KL}, ϵ_{LM}, and S in above, we have

$$\begin{aligned}
\mathcal{R}_n(\mathsf{MinimizePR}, \mathsf{PR}) &= N \cdot \left(1 + \left(\log \frac{1}{p'}\right)^{1/4} + \left(\log \frac{1}{p''}\right)^{1/2}\right) \cdot O\left((d_\Theta d_\Phi)^{1/2} \left(\frac{N_{\mathsf{KL}}}{N}\right)^{1/6}\right) \\
&= O\left(\left(1 + \left(\log \frac{1}{p'}\right)^{1/4} + \left(\log \frac{1}{p''}\right)^{1/2}\right) (d_\Theta + d_\Phi) N_{\mathsf{KL}}^{1/6} N^{5/6}\right)
\end{aligned}$$

We would like to ensure that this bound holds with probability $p > 0$. To that end, observe that the probabilistic terms are the high-probability bounds on $\mathcal{R}_S(\mathsf{LearnModel}, \mathsf{KL})$ and $\mathcal{R}_T(\mathsf{MinimizePR}, \mathsf{PR})$. By recalling Theorem 2 and Theorem 3, the probability that any of these bounds fails is at most

$$p' + T p_{\mathsf{LM}} = p' + T(p'' + S p_{\mathsf{KL}}) = p' + T p'' + S T p_{\mathsf{KL}}$$

for any $p', p'' > 0$. For a choice of $p' = p/3$, $p'' = p/3T$, and $p_{\mathsf{KL}} = \frac{p N_{\mathsf{KL}}}{3n}$, this is at most p as required. Finally, plugging these choices into the above regret bound yields

$$\begin{aligned}
\mathcal{R}_n(\mathsf{MinimizePR}, \mathsf{PR}) &= O\left(\left(1 + \left(\log \frac{1}{p'}\right)^{1/4} + \left(\log \frac{1}{p''}\right)^{1/2}\right) (d_\Theta + d_\Phi) N_{\mathsf{KL}}^{1/6} N^{5/6}\right) \\
&= O\left(\left(1 + \left(\log \frac{1}{p}\right)^{1/4} + \left(\log \frac{T}{p}\right)^{1/2}\right) (d_\Theta + d_\Phi) N_{\mathsf{KL}}^{1/6} N^{5/6}\right) \\
&= O\left(\left(1 + \sqrt{\log \frac{1}{p}}\right) (d_\Theta + d_\Phi) N_{\mathsf{KL}}^{1/6} N^{5/6} \sqrt{\log N}\right) \qquad (T \le N)
\end{aligned}$$

with probability at most p as required. □

G. Additional Related Work

In this section, we provide additional related work in performative prediction and a detailed comparison of our work and some closely related work.

Performative prediction is a new type of supervised learning problem in which the underlying data distribution shifts in response to the deployed model (Perdomo et al., 2020; Brown et al., 2022; Drusvyatskiy & Xiao, 2020; Izzo et al., 2021; Li & Wai, 2022; Maheshwari et al., 2022; Ray et al., 2022; Mofakhami et al., 2023). It is also called the *decision-dependent risk minimization* problem (Maheshwari et al., 2022; Li et al., 2022; Yuan & Gao, 2023). In particular, Perdomo et al. (2020) first propose the notion of the *performative risk* defined as $\mathsf{PR}(\theta) := \mathbb{E}_{z \sim \mathcal{D}(\theta)}[\ell(\theta; z)]$ where θ is the model parameter, and $\mathcal{D}(\theta)$ is the induced distribution due to the deployment of θ.

One of the major focuses of performative prediction is to find the optimal model θ_{OPT} which achieves the minimum performative prediction risk: $\theta_{\mathsf{OPT}} := \operatorname{argmin}_{\theta \in \Theta} \mathsf{PR}(\theta)$, or performative stable model θ_{ST}, which is optimal under its own induced distribution: $\theta_{\mathsf{ST}} := \operatorname{argmin}_{\theta \in \Theta} \mathbb{E}_{z \sim \mathcal{D}(\theta_{\mathsf{ST}})}[\ell(\theta; z)]$. In particular, one way to find a performative stable model θ_{ST} is to perform repeated retraining (Perdomo et al., 2020).

In order to get meaningful theoretical guarantees on any proposed algorithms, works in this field generally require particular assumptions on the mapping between the model parameter and its induced distribution (e.g., the smoothness of the mapping), or require multiple rounds of deployments and observing the corresponding induced distributions, which can be costly in practice (Jagadeesan et al., 2022; Mendler-Dünner et al., 2020). A few recent works are on finding performative optimal solutions without explicitly making the convexity assumption. For example, Dong & Ratliff (2021) does not explicitly convexity assumption, but they focus on optimization heuristics that are not guaranteed to minimize performative regret. below, we will provide the discussions for three of them.

In addition, minimizing the performative risk often requires knowing a specific model for the distribution map $\mathcal{D}(\cdot)$ that can be fit. To ensure performative risk minimization is tractable, one also requires imposing structural assumptions on the distribution map. For example, Izzo et al. makes parametric assumptions on $\mathcal{D}(\theta)$ and assumes that $\mathcal{D}(\theta)$ has a continuously differentiable density $p(z; \varphi(\theta))$, where $\varphi(\cdot) : \Theta \to \Phi$ represents the mapping from the model parameter space Θ to the data distribution parameter space Θ. Miller et al. (2021) assume the underlying data distribution follows a location family distribution, and then impose a *mixture dominance* assumption on the distribution map $\mathcal{D}(\cdot)$ from which it follows that $\mathsf{PR}(\theta)$ is convex; this again leads to a gradient-based optimization algorithm. Similar work include (Mendler-Dünner et al., 2020; Izzo et al., 2021; Drusvyatskiy & Xiao, 2020; Cutler et al., 2021), to name a few.

Comparison with Miller et al. (2021) Miller et al. (2021) identifies *mixture dominance condition* for any particular model parameter pairs under which the performative risk is convexity. In particular, they posit a simple distribution map in which $\varphi(\theta) = \phi_0 + M\theta$, where $M \in \mathbb{R}^{d_\Phi \times d_\Theta}$ and $\phi_0 \in \Phi$ is some "base" distribution parameter; in other words, they assume that the data population reacts to a model by shifting each of their features according to some linear transformation of the model parameter. Their algorithm for this special case works in two stages: first estimating φ_0 and M by deploying random models; then, once this distribution map has been accurately estimated, the performative loss is convex in θ, and can be optimized offline. The distribution map estimation takes $O(d_\Theta/\epsilon)$ samples to obtain an ϵ-suboptimal model.

Comparison with Jagadeesan et al. (2022) For example, closely related is a recent paper that proposes using the Lipschitz bandit approach to solve the performative prediction problem (Jagadeesan et al., 2022). The major differences between this work and their work are: first, we define the regret w.r.t N rather than w.r.t T, which is a more realistic measure in the performative prediction setting; second, their regret has exponential dependency on the "zooming dimension" d (which is roughly the model parameter d_Θ), while our dependency on the model and distribution dimensions are both linear.

Comparison with Maheshwari et al. (2022) Another closely related work is Maheshwari et al. (2022) uses zeroth-order methods for the convex-concave minimax problem. Specifically, they proposed to formulate the performative prediction problem as the *Wasserstein distributionally robust learning with decision-dependent data* problem, and further reduce it to a constrained finite-dimensional smooth convex-concave min-max problem, and propose a zeroth-order random reshuffling-based algorithm to solve the problem without assuming any other structure on the curvature of the min-max loss. Similar to ours, they also use the zeroth-order method to perform their optimization procedure; different from ours, they approach the performative prediction problem through the angle of robustness, which accounts for model misspecification in their analysis.

Other Aspects of performative prediction Also related are the recently developed lines of work on the *multiplayer* version of the performative prediction problem (Piliouras & Yu, 2022; Narang et al., 2022; Li et al., 2022; Foster et al., 2023). While existing strategic classification and performative prediction problems focus primarily on the interplay between a single learner and the population that reacts to the learner's actions, this line of work takes into account competition from multi-learners, and develop performatively stable equilibria and Nash equilibria of the game. Similarly, (Yuan & Gao, 2023) confront multiple interactive models in some dynamic environments. Another line of work is the economic aspects of performative prediction (Hardt et al., 2022; Mendler-Dünner et al.). From the optimization aspect, Wood & Dall'Anese (2022) focuses on the optimization aspect of finding the performative optimal point and offers an online stochastic primal-dual algorithm for tracking equilibrium trajectories. Also related is the recent development of the concept called induced domain adaptation (Chen et al., 2023), whose primary focus is to study the *transferability* of a particular model trained primarily on the source distribution and provide theoretical bounds on its performance on its induced distribution, which is helpful in estimating the effect of a given classifier when repeated retraining is unavailable.

Theoretical comparisons to some existing methods in the convex case When the problem reduces to a convex, differentiable Lipschitz case, our algorithm will be reduced to the convex case provided in Section 2.2 (the warm-up setting), which achieves a $\tilde{O}(\sqrt{dN})$ regret bound. This implies that our algorithm achieves a Δ-suboptimal model with $O(d/\Delta^2)$ samples (see Lemma 1).

Here, we compare the three papers that the reviewer mentioned:

- Izzo et al. (2021) focus on a single-distribution Gaussian distribution with a fixed variance setting while we cover a boarder range of settings. Their theoretical guarantee shows that the proposed method converges to a performative

optimal point as the number of iterations $T \approx \sigma^{-4/5}$ where σ bounds the output of PR from the PR of the optimal performative point. The sample required at each iteration is $O(1/\sigma^2 \log T)$.

- Miller et al. (2021) show that when the distribution maps $\mathcal{D}(\cdot)$ form a location-scale family and when the model dimension is $O(d)$, computing a Δ-suboptimal classifier requires $O(d/\delta)$ samples. We do not require these assumptions.
- Perdomo et al. (2020) focus on achieving a performative stable point while we focus on attaining a performative optimal point.

H. Plots For Empirical Results

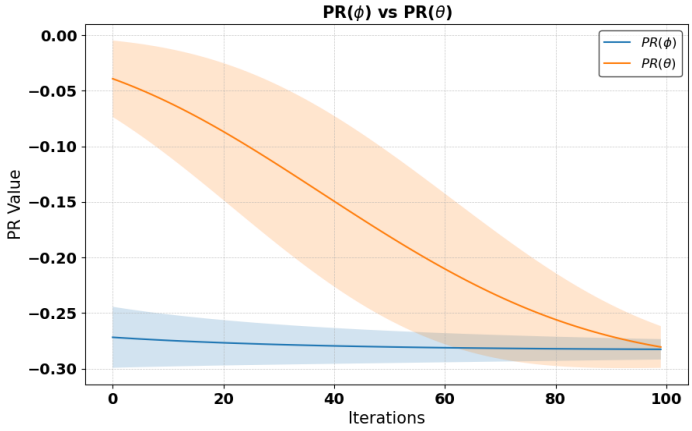

Figure 4. $\theta \in [0.2, 0.4]$

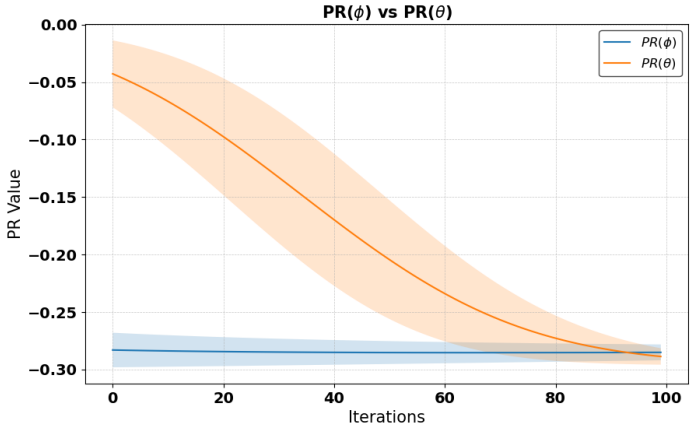

Figure 5. $\theta \in [0.4, 0.8]$

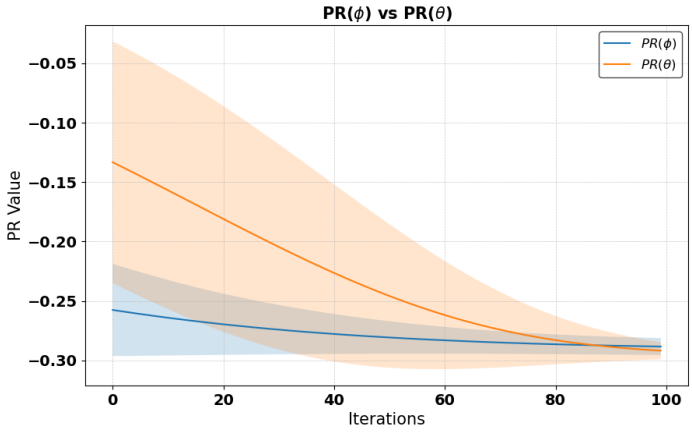

Figure 6. $\theta \in [0.4, 0.8]$

Figure 7. Empirical results comparing baseline method (zeroth-order optimization without reparametrization, orange curve) vs. our method (zeroth order optimization after reparametrization, (blue curve) based on Example 3.