SPRINT: Stochastic Performative Prediction With Variance Reduction

Anonymous authors
Paper under double-blind review

Abstract

Performative prediction (PP) provides a framework for learning in settings where model deployment influences the data distribution on which the model is trained. Unlike conventional learning under a fixed data distribution, PP involves model-induced distribution shifts that make it challenging to design algorithms converging to stable points, known as stationary performatively stable (SPS) solutions. Existing algorithms for finding SPS solutions, including repeated gradient descent (RGD) and greedy stochastic gradient descent (SGD-GD), have largely focused on strongly convex losses. Although recent work established an $\mathcal{O}(1/\sqrt{T})$ convergence guarantee for smooth non-convex losses, it relies on a bounded-variance assumption on stochastic gradients and converges only to a non-vanishing error neighborhood whose size scales with the variance. To address these limitations, we develop a variance-reduced stochastic optimization method for non-convex performative prediction. Our algorithm, **s**tochastic **p**erformative p**r**ediction w**i**th varia**n**ce reduc**t**ion (SPRINT), converges to an SPS solution at a rate of $\mathcal{O}(1/T)$. This guarantee removes the bounded-variance assumption on stochastic gradients and contains no variance-dependent residual error. Experiments on multiple real-world datasets with non-convex models demonstrate that SPRINTconverges faster and more stably than SGD-GD.

1 Introduction

Machine learning (ML) models are often trained under the assumption that both training and testing data are drawn independently from a fixed distribution. In many real-world applications, however, this assumption fails. When ML systems make predictions about humans or strategically adaptive agents, their deployment can influence the behavior of the population interacting with the system, thereby changing the data distribution on which the model is trained. This phenomenon, known as *performative effects* or *model-induced distribution shift* (Perdomo et al., 2021; Drusvyatskiy and Xiao, 2023), poses fundamental challenges to conventional learning paradigms. For example, in strategic classification, individuals may modify their features, such as resumes or financial profiles, to obtain more favorable predictions (Hardt et al., 2015); on digital platforms, users may adjust their engagement behavior based on their beliefs about how recommendation models operate (Zhang et al., 2019; Chi et al., 2022); and in adversarial settings such as spam detection, spammers continuously adapt their tactics to evade newly deployed filters. Understanding and mitigating such performative effects is therefore essential for building robust ML systems in adaptive environments.

To this end, Perdomo et al. (2021) introduced performative prediction (PP), an optimization framework in which the data distribution depends explicitly on the deployed model. In PP, the distribution $\mathcal{D}(\boldsymbol{\theta})$ is modeled as a mapping from the model parameters $\boldsymbol{\theta}$ to a space of data distributions, and the goal is to minimize the expected loss under this model-dependent distribution, known as the *performative risk* (PR):

$$\boldsymbol{\theta}^{\mathrm{PO}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathcal{V}(\boldsymbol{\theta}) \stackrel{\mathrm{def}}{=} \mathbb{E}_{z\sim\mathcal{D}(\boldsymbol{\theta})}[\ell(z;\boldsymbol{\theta})], \tag{1}$$

where $\ell(z;\boldsymbol{\theta})$ is the loss function and $z = (x, y)$ is sampled from the distribution $\mathcal{D}(\boldsymbol{\theta})$. The key challenge in PP is that $\boldsymbol{\theta}$ determines both the model and the data distribution. The minimizer $\boldsymbol{\theta}^{\mathrm{PO}}$ called the

performative optimal (PO) solution. However, because the distribution is itself a function of the variable $\boldsymbol{\theta}$ being optimized, directly finding $\boldsymbol{\theta}^{\mathrm{PO}}$ is often intractable (Perdomo et al., 2021; Izzo et al., 2021). To address this, much of the existing literature instead studies **performative stable** (PS) solution $\boldsymbol{\theta}^{\mathrm{PS}}$, which minimizes the **decoupled performative risk** $\mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}^{\mathrm{PS}})$ defined as follows:

$$\boldsymbol{\theta}^{\mathrm{PS}} = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}^{\mathrm{PS}}) \stackrel{\mathrm{def}}{=} \mathbb{E}_{z\sim\mathcal{D}(\boldsymbol{\theta}^{\mathrm{PS}})}[\ell(z;\boldsymbol{\theta})]. \tag{2}$$

Unlike the performative risk $\mathcal{V}(\boldsymbol{\theta})$, where the data distribution $\mathcal{D}(\boldsymbol{\theta})$ depends on the optimization variable, $\mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}^{\mathrm{PS}})$ decouples the two: the data distribution is induced by a fixed parameter $\boldsymbol{\theta}^{\mathrm{PS}}$, while optimization is performed over $\boldsymbol{\theta}$. This decoupling enables iterative schemes for finding $\boldsymbol{\theta}^{\mathrm{PS}}$ via *repeated optimization*: at each iteration, the learner updates $\boldsymbol{\theta}_{t+1}$ by minimizing the risk under the distribution induced by the previously deployed model $\mathcal{D}(\boldsymbol{\theta}_t)$. If this procedure converges, the iterates approach a PS solution. Examples of this include repeated risk minimization (Perdomo et al., 2021), repeated gradient descent (Perdomo et al., 2021), and stochastic gradient descent-greedy deploy (SGD-GD) (Mendler-Dünner et al., 2020).

In practice, computing full-batch gradients at every deployment step can be computationally expensive, especially when the interacting population is large (Perdomo et al., 2021). This motivates stochastic methods, which replace full-batch updates with stochastic gradient estimates computed from samples drawn from the current performative distribution. In this work, we focus on SGD-GD for two main reasons. First, unlike RGD (Perdomo et al., 2021), which requires computing a full gradient at each iteration, SGD-GD uses only a single stochastic gradient sample per update and is therefore more scalable to large populations. Second, unlike derivative-free methods (Ray et al., 2022; Liu et al., 2023) and performative gradient descent (Izzo et al., 2021), SGD-GD does not require explicit knowledge of the distribution map $\mathcal{D}(\theta)$. In general, the SGD-GD update can be formalized as follows.

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \gamma_{t+1}\nabla\ell(z_{t+1};\boldsymbol{\theta}_t), \quad z_{t+1}\sim\mathcal{D}(\boldsymbol{\theta}_t). \tag{3}$$

Prior work has shown that Eqn. 3 can converge to a unique PS solution (Mendler-Dünner et al., 2020), but under two strong assumptions: (i) the loss function ℓ is strongly convex and smooth in $\boldsymbol{\theta}$; and (ii) the sensitivity of the distribution map $\mathcal{D}$, measured by the Wasserstein-1 distance, is bounded. However, the strong convexity requirement substantially limits the applicability of these results. Li and Wai (2024) relaxed this assumption and established the first convergence guarantees for SGD-GD under smooth non-convex losses, showing convergence to a stationary performatively stable (SPS) solution (cf. Def. 1.1).

Definition 1.1 (δ-Stationary Performative Stable Solution (Li and Wai, 2024))**.** For a given $\delta \geq 0$, the vector $\boldsymbol{\theta}_{\delta\text{-SPS}} \in \mathbb{R}^d$ is said to be a δ-stationary performative stable (δ-SPS) solution of Eqn. 2 if

$$\|\nabla\mathcal{J}(\boldsymbol{\theta}_{\delta\text{-SPS}};\boldsymbol{\theta}_{\delta\text{-SPS}})\|^2 = \left\|\mathbb{E}_{z\sim\mathcal{D}(\boldsymbol{\theta}_{\delta\text{-SPS}})}\left[\nabla\ell(z;\boldsymbol{\theta}_{\delta\text{-SPS}})\right]\right\|^2 \leq \delta.$$

The parameter δ in Def. 1.1 measures the stability of a solution: when $\delta = 0$, $\boldsymbol{\theta}_{\delta\text{-SPS}}$ recovers the PS solution $\boldsymbol{\theta}^{\mathrm{PS}}$. As shown by Li and Wai (2024), SGD-GD can converge to $\boldsymbol{\theta}_{\delta\text{-SPS}}$ at a rate of $\mathcal{O}(1/\sqrt{T})$ for smooth non-convex loss function ℓ. However, this result relies on an additional assumption on **bounded-variance of stochastic gradient estimates**. Moreover, the resulting convergence bound contains a *non-vanishing* error neighborhood that *scales with the variance*. This motivates the following question:

(Q): Can we control the variance of stochastic gradient estimates in PP to design an SGD-GD-type algorithm that converges to $\boldsymbol{\theta}_{\delta\text{-SPS}}$ with (i) a faster rate and (ii) an error neighborhood *independent* of the gradient variance? If yes, how?

In conventional (non-PP) stochastic optimization, variance reduction has proved effective for accelerating SGD and improving convergence guarantees. Representative methods include stochastic average gradient (SAG) (Roux et al., 2012), stochastic average gradient augmented (SAGA) (Defazio et al., 2014), stochastic variance reduced gradient (SVRG) (Reddi et al., 2016), and stochastic path-integrated differential estimator (SPIDER) (Fang et al., 2018). However, whether such techniques can be adapted to performative prediction remains largely *unknown*. This paper addresses this gap by developing a variance-reduced SGD-GD-type framework for PP.

As a first step, we propose **s**tochastic **p**erformative p**r**ediction w**i**th varia**n**ce reduc**t**ion (SPRINT), a variance-reduced stochastic optimization framework for PP inspired by stochastic variance reduced gradient (SVRG) (Reddi et al., 2016). In standard non-PP settings, SVRG periodically computes the full gradient at a past iterate (referred to as the snapshot point) and uses it to correct the stochastic gradient estimator in subsequent iterations. Specifically, SVRG combines the current stochastic gradient with a gradient-difference correction between the current iterate and the snapshot point, together with the stored full gradient at the snapshot point. This correction reduces the variance of stochastic updates relative to vanilla SGD, particularly when the current iterate remains close to the snapshot point.

Building on this idea, SPRINT divides the SGD-GD updates into multiple epochs. At the beginning of each epoch s, the algorithm stores a snapshot point $\widetilde{\boldsymbol{\theta}}^s$ and computes the corresponding full gradient of the decoupled performative risk $\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)$. During the inner iterations, this snapshot information is used to construct variance-reduced stochastic gradient estimators for updating the iterates $\boldsymbol{\theta}_k^s$.

We emphasize that SPRINT is **not** a straightforward extension of SVRG to the PP setting. Unlike conventional stochastic optimization, the performative nature of the problem introduces two unique challenges for both algorithm design and theoretical analysis. First, because performative effects continuously modify the data distribution across iterations, the gradient of the decoupled performative risk also evolves over time. Consequently, the full-gradient snapshot computed at $\widetilde{\boldsymbol{\theta}}^s$ under a previous distribution becomes a biased estimate of the current gradient, even when evaluated at the same parameter value. Second, the evolving data distribution introduces additional analytical difficulties absent in standard SVRG analysis. In particular, the update from $\boldsymbol{\theta}_k^s$ to $\boldsymbol{\theta}_{k+1}^s$ induces extra distribution shifts, making it substantially more challenging to construct a suitable Lyapunov function and establish a negative Lyapunov drift for proving convergence.

The main contribution of this paper is to overcome the above technical challenges and establish convergence of the proposed SPRINT method to an SPS solution with a fast *variance-independent* convergence rate. Our main results and key contributions are summarized as follows:

- We propose the SPRINT algorithm to reduce the variance of SGD-GD-based approaches in non-convex PP settings. Compared to existing methods, SPRINT converges to an SPS solution at an *accelerated* rate of $\mathcal{O}(1/T)$ and, more importantly, achieves an error neighborhood that is **independent** of the stochastic gradient variance. To the best of our knowledge, these results are novel in the literature.
- To establish the $\mathcal{O}(1/T)$ convergence rate of SPRINT , we develop a series of new Lyapunov function construction techniques, which may be of independent interest in the PP literature. Moreover, we derive the incremental first-order oracle (IFO) complexity (i.e., the number of stochastic gradient evaluations required to achieve a $\mathcal{O}(\delta)$-accurate SPS solution (Reddi et al., 2016)).
- We conduct extensive numerical experiments to validate our theoretical results on three real-world datasets. In addition to credit data (Kaggle, 2012) and MNIST (Deng, 2012), which are commonly used in PP evaluations, we include additional experiments training MLP/CNN models using both SGD-GD-based and SPRINT methods on the full CIFAR-10 (Krizhevsky and Hinton, 2009) dataset. These experiments demonstrate and further support the practical effectiveness of optimization-based approaches for PP.

The remainder of this paper is organized as follows: In Sec. 2, we review related work and present the problem formulation, along with a discussion of prior results in Sec. 3. We then introduce the SPRINT algorithm in Sec. 4 and establish its convergence and IFO complexity in Sec. 5. Finally, we present numerical results in Sec. 6 and conclude the paper in Sec. 7.

2 Related work

We review four closely related lines of work: (i) algorithms for finding PS/SPS solutions in performative prediction; (ii) alternative solution concepts and metrics in PP; (iii) variance-reduction techniques for SGD-based optimization; (iv) connections between PP, reinforcement learning, and optimal control.

1) Finding PS/SPS solutions for PP. Performative prediction (PP) was first introduced by Perdomo et al. (2021) as an optimization framework for learning under endogenous data distribution shifts. They

proposed *repeated risk minimization* (RRM), an iterative optimization procedure for finding performatively stable (PS) solutions, and established bounds on the distance between PS and performatively optimal (PO) solutions. They also studied repeated gradient descent (RGD), which updates the model by taking full-gradient steps under the currently induced distribution. However, RGD requires access to a full gradient at every iteration, which can be computationally expensive in large-scale settings. To improve scalability, Mendler-Dünner et al. (2020) developed the first stochastic algorithm for finding PS solutions in an online setting. Later, Mendler-Dünner et al. (2022) established convergence rates for greedy and lazy deployment schemes under smoothness and strong convexity assumptions. Other works have extended PP to more structured learning settings. For example, Mofakhami et al. (2023) studied neural-network training in PP settings, assuming a predictor of the form $\hat{y} = f_{\boldsymbol{\theta}}(x)$, a loss $\ell(\hat{y}, y)$ that is strongly convex in $\hat{y}$, and a distribution map $D(\hat{y})$ that is ϵ-sensitive with respect to predictions. From a complexity-theoretic perspective, Anagnostides et al. (2026) showed that finding a performatively stable point is PPAD-complete even for Lipschitz losses, highlighting the need for efficient stochastic methods that converge to approximate SPS solutions.

Most closely related to our work, Li and Wai (2024) provided the first convergence guarantee for PP with smooth non-convex losses without imposing a particular solution structure. Their result shows that SGD-GD converges to a stationary performatively stable (SPS) solution. However, the guarantee relies on a bounded-variance assumption on stochastic gradient estimates and yields a non-vanishing error neighborhood that scales with the variance. Our work overcomes this limitation by introducing a variance-reduced SGD-GD-type method that accelerates convergence while removing variance-dependent residual error.

2) Alternative solution concepts and metrics in PP. Beyond PS and SPS solutions, several works have studied alternative solution concepts for PP. One line of work directly optimizes the performative risk (PR) to find PO solutions. For instance, Miller et al. (2021) studied direct optimization of PR under restricted classes of distribution maps, while Ray et al. (2022); Liu et al. (2023) used derivative-free optimization methods to find PO solutions. Izzo et al. (2021) proposed performative gradient descent for finding PO solutions under a convex PR assumption; however, convexity of the performative risk can be difficult to verify, even when the loss itself is strongly convex. Other works have studied PP under weak convexity assumptions (Zhu et al., 2023; Zheng et al., 2024) or in state-dependent settings, where algorithms are designed to converge to PS solutions (Brown et al., 2022; Li and Wai, 2022).

In addition to optimization-based solution concepts, recent work has examined broader statistical and societal impacts of performative effects. For example, Tsoy et al. (2025) analyzed the impact of performative risk minimization on binary random variables, showing that PRM can induce non-vanishing prediction bias and distributional mean shift. Other studies have investigated performative prediction through the lens of fairness (Jin et al., 2024), social welfare (Kim and Perdomo, 2022), and privacy (Li et al., 2025). These works highlight the broader importance of understanding model-induced distribution shifts, while our work focuses specifically on improving stochastic optimization guarantees for finding SPS solutions.

3) ariance-reduction techniques for SGD-based optimization. Variance reduction (VR) has been extensively studied in conventional stochastic optimization as a means of accelerating SGD and improving convergence guarantees (Roux et al., 2012; Defazio et al., 2014; Reddi et al., 2016; Nguyen et al., 2017; Fang et al., 2018). Representative methods include stochastic average gradient (SAG) (Roux et al., 2012) and stochastic average gradient augmented (SAGA) (Defazio et al., 2014), which maintain and update a table of per-sample gradients to reduce the variance of stochastic updates. While effective, these methods can incur high memory costs in large-scale regimes.

Stochastic variance-reduced gradient (SVRG) (Reddi et al., 2016) addresses this issue by avoiding the storage of per-sample gradients. Instead, SVRG periodically computes a full gradient at a snapshot point and uses it to construct variance-reduced stochastic updates throughout an epoch. Under strong convexity, SVRG achieves linear convergence with modest memory requirements. Subsequent methods, including stochastic recursive gradient methods such as SARAH (Nguyen et al., 2017) and stochastic path-integrated differential estimator (SPIDER) (Fang et al., 2018), further improve variance reduction by using recursive gradient estimators based on fresher iterates, achieving optimal convergence rates in terms of stationary gap and dataset-size dependence.

However, existing VR techniques are designed for standard stochastic optimization under fixed data distributions. They do not directly apply to PP, where the data distribution changes as the model is updated. This model-induced distribution shift makes it nontrivial to construct unbiased or low-variance gradient estimators and to establish convergence guarantees. To the best of our knowledge, variance-reduced stochastic optimization for performative prediction remains largely underexplored. Our work addresses this gap by developing a variance-reduced SGD-GD-type framework tailored to PP.

4) Connections between PP, reinforcement learning, and optimal control. Performative prediction is also related to reinforcement learning (RL) and optimal control (OC), as all three frameworks involve some form of decision-dependent data or state evolution. In RL (Kaelbling et al., 1996), a policy π is optimized to maximize an expected reward objective of the form $J(\pi) = \mathbb{E}_{\tau\sim p_\pi}[R(\tau)]$, where the trajectory distribution p_π depends on the policy. However, RL is typically reward-driven and focuses on sequential decision-making, rather than on finding an equilibrium between a deployed model and the distribution it induces. For convergence analyses of various optimization algorithms in RL, we refer readers to Xiao (2022); Beggs (2005); Agarwal and Agarwal (2015); Yuan et al. (2022).

Optimal control (Todorov et al., 2006) studies dynamic systems governed by temporal state transitions, such as $x_{t+1} = f(x_t, u_t)$, and seeks to minimize cumulative costs of the form $\sum_t c(x_t, u_t)$. The dependency in OC is primarily temporal and state-based, whereas PP focuses on model-dependent distribution shift: the deployed model changes the data distribution on which future learning is performed. Thus, while PP shares conceptual connections with RL and OC, its equilibrium-based optimization structure and model-induced distribution shifts create distinct algorithmic and analytical challenges.

3 Preliminaries

Following the convention in the variance reduction literature (Reddi et al., 2016; Roux et al., 2012), we assume the full population consists of n samples $\{z_1, ..., z_n\} \subset \mathcal{Z}$, [1] and that the ML model is parameterized by $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d$. The expected gradient of the decoupled performative risk over the full population is

$$\nabla \mathcal{J}(\boldsymbol{\theta}; \boldsymbol{\theta}) = \mathbb{E}_{z\sim\mathcal{D}(\boldsymbol{\theta})}[\nabla \ell(z; \boldsymbol{\theta})] = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell(z_i; \boldsymbol{\theta}),$$

where $\nabla \ell(z; \boldsymbol{\theta})$ denotes the gradient of $\ell(z; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

We adopt the *non-convex* problem setup of Li and Wai (2024), in contrast to most prior PP work that assumes loss function ℓ is strongly convex (Mendler-Dünner et al., 2020; Izzo et al., 2021; Mofakhami et al., 2023). To set up the background, we first present the technical assumptions and then the convergence result of Li and Wai (2024). Throughout the paper, we use $\|\cdot\|$ to denote ℓ_2 norm.

Assumption 3.1 (Smoothness and lower bound of the loss)**.** There exist constants $L, L_0 \geq 0$ such that

$$\|\nabla \ell(z; \boldsymbol{\theta}) - \nabla \ell(z; \boldsymbol{\theta'})\| \leq L\|\boldsymbol{\theta} - \boldsymbol{\theta'}\|, \quad \forall \boldsymbol{\theta}, \boldsymbol{\theta'} \in \mathbb{R}^d$$
$$\|\nabla \ell(z; \boldsymbol{\theta}) - \nabla \ell(z'; \boldsymbol{\theta})\| \leq L_0\|z - z'\|, \quad \forall z, z' \in \mathcal{Z}$$

Moreover, there exists a constant $\ell^\star > -\infty$ such that $\ell(z; \boldsymbol{\theta}) \geq \ell^\star$ for all $\boldsymbol{\theta}, z$.

Definition 3.2 (ϵ-sensitivity)**.** Distribution map $\mathcal{D}(\theta)$ is ϵ-sensitive if there exists $\epsilon \geq 0$ such that

$$W_1(\mathcal{D}(\boldsymbol{\theta}), \mathcal{D}(\boldsymbol{\theta'})) \leq \epsilon\|\boldsymbol{\theta} - \boldsymbol{\theta'}\|, \ \forall \boldsymbol{\theta}, \boldsymbol{\theta'} \in \mathbb{R}^d$$

where $W_1(\cdot, \cdot)$ is the Wasserstein-1 distance (Li and Wai, 2024).

Assumption 3.3 (Lipschitzness of the loss)**.** The loss function ℓ is L_1-Lipschitz in z:

$$|\ell(z; \boldsymbol{\theta}) - \ell(z'; \boldsymbol{\theta})| \leq L_1\|z - z'\|, \quad \forall z, z' \in \mathcal{Z}, \boldsymbol{\theta} \in \mathbb{R}^d.$$

[1] This finite-population setting is standard in the variance reduction literature (Roux et al., 2012; Reddi et al., 2016; Nguyen et al., 2017) and is natural in PP, where typical applications such as credit scoring (Perdomo et al., 2021) and college admission (Xie et al., 2024)) often involve a finite population.

Literature	Loss function ℓ	Convergence rate	Error neighborhood	Comments
Mendler-Dünner et al. (2020)	Strongly convex	$\mathcal{O}(\frac{1}{T})$	None	No error due to strong convexity
Li and Wai (2024)	Nonconvex	$\mathcal{O}(\frac{1}{\sqrt{T}})$	$\mathcal{O}(\sigma_0\epsilon + \sigma_1^2\epsilon^2)$	Variance-dependent error neighborhood
This work: SPRINT	Nonconvex	$\mathcal{O}(\frac{1}{T})$	$\mathcal{O}(\epsilon^2 + \epsilon^4)$	**Variance-independent** error neighborhood

Table 1: SPRINT compared to prior SGD-GD-based methods for PP. Since our main objective is to improve existing SGD-GD-based PP approaches, we do not include algorithms targeting PO solutions (Izzo et al., 2021; Miller et al., 2021) or those based on repeated risk minimization (RRM) (Perdomo et al., 2021; Mofakhami et al., 2023); we refer readers to Table 1 of Li and Wai (2024) for a detailed comparison with these methods.

Assumption 3.4 (Variance of stochastic gradient estimates)**.** The stochastic gradient is unbiased, i.e., $\nabla\mathcal{J}(\boldsymbol{\theta_1};\boldsymbol{\theta_2}) = \mathbb{E}_{Z\sim\mathcal{D}(\boldsymbol{\theta_2})}[\nabla\ell(Z;\boldsymbol{\theta_1})]$. Moreover, there exist constants $\sigma_0, \sigma_1 \geq 0$, such that

$$\mathbb{E}_{Z\sim\mathcal{D}(\boldsymbol{\theta_2})} \left\|\nabla\ell(Z;\boldsymbol{\theta}_1) - \nabla\mathcal{J}(\boldsymbol{\theta}_1;\boldsymbol{\theta}_2)\right\|^2 \leq \sigma_0^2 + \sigma_1^2 \left\|\nabla\mathcal{J}(\boldsymbol{\theta}_1;\boldsymbol{\theta}_2)\right\|^2$$

Note that smoothness (Assumption 3.1) and ϵ-sensitivity (Def. 3.2) are standard in the PP optimization literature (Perdomo et al., 2021; Mendler-Dünner et al., 2020), while Lipschitzness of loss (Assumption 3.3) and the bounded-variance condition (Assumption 3.4) are specific to non-convex PP but remain common in optimization. Note that Assumption 3.3 is only about Lipschitzness with respect to z and does not imply the bounded-variance assumption.

Under above assumptions, Li and Wai (2024) proved the following convergence result:

Lemma 3.5 (Convergence of SGD-GD (Li and Wai, 2024))**.** *For the SGD-GD iterates given in Eqn. 3, the following holds for any $T > 1$:*

$$\sum_{t=0}^{T-1} \frac{\gamma_{t+1}}{4}\mathbb{E}\left[\|\nabla J(\boldsymbol{\theta}_t;\boldsymbol{\theta}_t)\|^2\right] \leq \Delta_0 + \frac{L}{2}\sigma_0^2 \sum_{t=0}^{T-1}\gamma_{t+1}^2 + L_1\epsilon\left(\sigma_0 + (1+\sigma_1^2)L_1\epsilon\right)\sum_{t=0}^{T-1}\gamma_{t+1}, \tag{4}$$

where γ_t is the learning rate at iteration t and $\Delta_0 = \mathcal{J}(\boldsymbol{\theta_0};\boldsymbol{\theta_0}) - \ell_*$ is an upper bound of the initial optimality gap between the initial performative risk and the optimal performative risk ℓ_*. If one sets $\gamma_t = 1/\sqrt{T}$ for all t, the bound in 4 can be reduced to:

$$\mathbb{E}\left[\|\nabla J(\boldsymbol{\theta}_\tau;\boldsymbol{\theta}_\tau)\|^2\right] \leq 4 \underbrace{\left(\Delta_0 + \frac{L}{2}\sigma_0^2\right)}_{\text{Variance-Dependent}} \frac{1}{\sqrt{T}} + 4L_1\epsilon \underbrace{\left(\sigma_0 + (1+\sigma_1^2)L_1\epsilon\right)\}}_{\text{Variance-Dependent}}. \tag{5}$$

Lemma 3.5 shows that SGD-GD converges at a rate of $\mathcal{O}(1/\sqrt{T})$ to an error neighborhood of size $\mathcal{O}(4L_1\epsilon(\sigma_0 + (1+\sigma_1^2)L_1\epsilon)$ [2]. However, this error neighborhood is directly *influenced by the variance* of the stochastic gradient estimate. Additionally, the convergence rate is affected by the second term on the right-hand side of Eqn. 4, which is also *variance-dependent*. These limitations motivate us to propose a new algorithm for **non-convex** PP that achieves faster convergence to SPS solutions and to develop new variance reduction techniques that enable the SGD-GD-based PP approach to be **variance-independent**.

4 The proposed SPRINT algorithm

To address the limitations of Lemma 3.5, we propose SPRINT, a new algorithm for non-convex PP inspired by stochastic variance reduced gradient (SVRG) (Reddi et al., 2016). As shown in Alg. 1, the total T update iterations are divided into S epochs, each consisting of m iterations. At the beginning of each epoch $s+1$, the algorithm sets a snapshot point $\widetilde{\boldsymbol{\theta}}^s$ and computes the corresponding full gradient of the decoupled performative risk, $\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)$ (Line 4). During each inner iteration k of the epoch, the algorithm constructs

[2]This rate also holds in the finite-population setting.

a variance-reduced estimator $\boldsymbol{v}_k^{s+1}$ (Line 7) by subtracting the control variate $\nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s)$ from the current stochastic gradient $\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1})$ and adding back the full gradient $\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)$ at the snapshot.

Note that, unlike variance reduction in the non-PP setting, $\boldsymbol{v}_k^{s+1}$ is *not* an unbiased estimator of $\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1}) = \mathbb{E}_{z\sim\mathcal{D}(\boldsymbol{\theta}_k^{s+1})}[\nabla\ell(z;\boldsymbol{\theta}_k^{s+1})]$ in PP setting. This is because the snapshot gradient is computed under $\mathcal{D}(\widetilde{\boldsymbol{\theta}}^s)$, whereas the current stochastic gradient is sampled from $\mathcal{D}(\boldsymbol{\theta}_k^{s+1})$, and generally $\mathcal{D}(\widetilde{\boldsymbol{\theta}}^s) \neq \mathcal{D}(\boldsymbol{\theta}_k^{s+1})$. This **new bias**, absent from the non-PP setting, significantly complicates the convergence analysis, as detailed in Sec. 5. Table 1 compares SPRINT with existing SGD-GD-based PP methods. Notably, only Li and Wai (2024) and our work address the non-convex PP setting. Among the two, SPRINT achieves a faster convergence rate and yields an error neighborhood independent of the stochastic-gradient variance, as established in Sec. 5.

Algorithm 1 Proposed Algorithm: Stochastic Performative Prediction with Variance Reduction (SPRINT)

Require: Number of total rounds T, number of iterations m each epoch, learning rates $\gamma_0, \cdots, \gamma_{m-1}$, initialization $\widetilde{\boldsymbol{\theta}}^0 = \boldsymbol{\theta}_m^0 = \boldsymbol{\theta}_0$.

1: $S = \lceil T/m \rceil$
2: **for** $s = 0$ to $S-1$ **do**
3: $\quad\boldsymbol{\theta}_0^{s+1} = \boldsymbol{\theta}_m^s = \widetilde{\boldsymbol{\theta}}^s$
4: $\quad\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s) = \frac{1}{n}\sum_{i=1}^n[\nabla\ell(z_i;\widetilde{\boldsymbol{\theta}}^s)]$ where each $z_i \sim \mathcal{D}(\widetilde{\boldsymbol{\theta}}^s)$
5: $\quad$**for** $k = 0$ to $m-1$ **do**
6: $\quad\quad$Pick a sample z_{i_k} where $z_{i_k} \sim \mathcal{D}(\boldsymbol{\theta}_k^{s+1})$
7: $\quad\quad\boldsymbol{v}_k^{s+1} = \nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1}) - \nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s) + \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)$
8: $\quad\quad\boldsymbol{\theta}_{k+1}^{s+1} = \boldsymbol{\theta}_k^{s+1} - \gamma_k\boldsymbol{v}_k^{s+1}$
9: $\quad$**end for**
10: $\quad\widetilde{\boldsymbol{\theta}}^{s+1} = \boldsymbol{\theta}_m^{s+1}$
11: **end for**

Output: $\boldsymbol{\theta}_m^S$

5 Convergence analysis of SPRINT

In this section, we analyze the convergence of SPRINT .

Assumptions. Our analysis relies only on smoothness of the loss (Assumption 3.1), ϵ-sensitivity of the distribution map (Def. 3.2), and Lipschitzness of the loss (Assumption 3.3). Crucially, unlike Li and Wai (2024), we **do not** require the variance of the stochastic gradient estimate is bounded (Assumption 3.4).

Overview of the proof. Following Reddi et al. (2016) for SVRG in the non-PP setting, we establish convergence of SPRINT by constructing a suitable *Lyapunov function* and showing that the corresponding function sequence decreases over iterations. However, the analysis of (Reddi et al., 2016) does not extend directly to PP: in the non-PP setting, the loss $\ell(z;\boldsymbol{\theta})$ depends only on the current parameter $\boldsymbol{\theta}$, whereas the decoupled performative risk $\mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}') = \mathbb{E}_{z\sim\mathcal{D}(\boldsymbol{\theta}')}[\ell(z;\boldsymbol{\theta})]$ depends on two arguments: $\boldsymbol{\theta}$, which determines the model prediction, and $\boldsymbol{\theta}'$, which controls the data distribution. Because each parameter update simultaneously shifts the data distribution, bounding consecutive Lyapunov terms is substantially harder and calls for new analysis techniques.

We overcome these challenges and establish the SPS convergence of SPRINT in three steps: (i) **Construct an intermediate function sequence** $R_k^{s+1}(\boldsymbol{\theta})$ (Lemma 5.1), in which consecutive terms differ only in the *second* argument of decoupled performative risk $\mathcal{J}$ and show that $R_k^{s+1}(\boldsymbol{\theta}_k^{s+1})$ is smaller than $R_{k+1}^{s+1}(\boldsymbol{\theta}_k^{s+1})$; (ii) **Construct the final Lyapunov function** $\widetilde{R}_k^{s+1}$ (Lemma 5.3) in which consecutive terms differ in *both* arguments of $\mathcal{J}$; (iii) **Establish convergence** by proving that $\widetilde{R}_k^{s+1}$ is decreasing, leading to the main convergence result (Theorem 5.4) and the IFO complexity bound (Corollary 5.5).

Step 1: Construct an intermediate function sequence. For the $(s+1)$-th epoch, we identify an intermediate function evaluated at the iterations k and $k+1$ as:

$$R_k^{s+1}(\boldsymbol{\theta}) \triangleq \mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}) + c_k\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2\right], \quad R_{k+1}^{s+1}(\boldsymbol{\theta}) \triangleq \mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}) + c_{k+1}\|\boldsymbol{\theta}_{k+1}^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2\right].$$

Then, the following Lemma 5.1 characterizes the relation between $R_k^{s+1}(\boldsymbol{\theta}_k^{s+1})$ and $R_{k+1}^{s+1}(\boldsymbol{\theta}_k^{s+1})$:

Lemma 5.1. *Let β_k be some positive constant, and let c_k, c_{k+1} be constants defined in intermediate functions R_k^{s+1}, R_{k+1}^{s+1} that satisfy the following:*

$$c_k - \frac{1}{2}\gamma_k = c_{k+1}(1 + \gamma_k\beta_k + 2L_0\epsilon\gamma_k) + (2L^2 + L_0^2\epsilon^2)(L\gamma_k^2 + 2c_{k+1}\gamma_k^2) + \frac{\beta_k}{2}L_0\epsilon\gamma_k \tag{6}$$

Then, we have

$$R_{k+1}^{s+1}(\boldsymbol{\theta}_k^{s+1}) \leq R_k^{s+1}(\boldsymbol{\theta}_k^{s+1}) - \Gamma_k\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right] - \frac{1}{2}\gamma_k\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2, \tag{7}$$

where $\Gamma_k = \left(\gamma_k - \frac{c_{k+1}\gamma_k + \frac{1}{2}L_0\epsilon\gamma_k}{\beta_k} - 2L\gamma_k^2 - 4c_{k+1}\gamma_k^2\right)$.

Note that $R_{k+1}^{s+1}(\boldsymbol{\theta}_k^{s+1})$ includes a term $\mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_k^{s+1})]$, which represents the expected performative risk when the model parameter is $\boldsymbol{\theta}_{k+1}^{s+1}$ but the data distribution is induced by $\boldsymbol{\theta}_k^{s+1}$. This enables us to focus on bounding the influence of $\boldsymbol{\theta}$ through the first argument of $\mathcal{J}$, similar to analyses in the non-PP settings. However, since the snapshot full gradient $\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)$ is taken under the *previous* distribution, we must additionally bound the effect of the distribution shift from $\mathcal{D}(\widetilde{\boldsymbol{\theta}}^s)$ to $\mathcal{D}(\boldsymbol{\theta}_k^{s+1})$. To this end, we provide the following auxiliary lemma.

Lemma 5.2 (Li and Wai (2024))**.** *Under Assumption 3.3 and ϵ-sensitivity in Def. 3.2, we have:*

$$|\mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}_1) - \mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}_2)| \leq L_1\epsilon\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|, \quad \forall\boldsymbol{\theta}.$$

The proof is given in App. B of Li and Wai (2024). To prove Lemma 5.1, we first apply the smoothness conditions of ℓ on $\mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_k^{s+1})]$ and leverage Lemma 5.2 to bound $\|\boldsymbol{v}_k^{s+1}\|$ and $\|\boldsymbol{v}_k^{s+1}\|^2$. The full proof is deferred to App. B.

Step 2: Construct the final Lyapunov function. We then construct the final Lyapunov function: at epoch $s+1$, we define Lyapunov function evaluated at iterations k and $k+1$ as:

$$\widetilde{R}_k^{s+1} \triangleq \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})] + c_k\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2, \quad \widetilde{R}_{k+1}^{s+1} \triangleq \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_{k+1}^{s+1})] + c_{k+1}\|\boldsymbol{\theta}_{k+1}^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2.$$

The following Lemma 5.3 characterizes the relation between $\widetilde{R}_{k+1}^s$ and $\widetilde{R}_{k+1}^{s+1}$:

Lemma 5.3. *Let $c_k, c_{k+1}, \beta_k, \Gamma_k$ be defined the same as in Lemma 5.1, we have:*

$$\widetilde{R}_{k+1}^{s+1} \leq \widetilde{R}_k^{s+1} + (L_0^2L_1^2\epsilon^4 + 2L_1^2\epsilon^2L^2 + 2L_1^2\epsilon^2)\gamma_k - \left(\Gamma_k - \frac{\gamma_k}{4}\right)\cdot\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right] \tag{8}$$

Lemma 5.3 can be proved by leveraging Lemmas 5.1 and 5.2. The proof also establishes a bound on the difference between $\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})$ and $\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_{k+1}^{s+1})$. The full proof is provided in App. C.

Step 3: Establish the convergence of SPRINT Building on Lemma 5.3, we are ready to show the convergence of SPRINT algorithm.

Theorem 5.4 (Convergence to an SPS point)**.** *Consider* SPRINT *(Algorithm 1) run for T total iterations with $S = \lceil T/m \rceil$ epochs. Set $c_m = 0$ in the Lyapunov function for the final iteration of each epoch. Then*

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{k=0}^{m-1}\mathbb{E}\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2 \leq \frac{\Delta_0}{T\Gamma} + \Delta_1, \tag{9}$$

where terms Δ_0 and Δ_1 are defined as

$$\Delta_0 \triangleq \mathcal{J}(\boldsymbol{\theta}_0^0;\boldsymbol{\theta}_0^0) - \ell_*; \quad \Delta_1 \triangleq \frac{(L_0^2L_1^2\epsilon^4 + 2L_1^2\epsilon^2L^2 + 2L_1^2\epsilon^2)\sum_{s=0}^{S}\sum_{k=0}^{m}\gamma_k}{T\Gamma}$$

and $\Gamma > 0$ is a constant lower bound on $\Gamma_k - \frac{\gamma_k}{4}$, whose existence is guaranteed. Consequently, Algorithm 1 converges at rate $\mathcal{O}(1/T)$ to a variance-independent error neighborhood $\Delta_1 = \mathcal{O}(\epsilon^2 + \epsilon^4)$.

Eqn. 9 is established by summing the bound in Lemma 5.3 over k and s and lower bounding $\Gamma_k - \frac{1}{4}\gamma_k$ by Γ; the full proof is in App. D. Theorem 5.4 improves on previous results in three aspects: (i) the convergence rate $\mathcal{O}(1/T)$ is faster than the $\mathcal{O}(1/\sqrt{T})$ rate of Li and Wai (2024); (ii) we eliminate Assumption 3.4, and the error neighborhood Δ_1 **no longer depends** on the variance of the stochastic gradient; and (iii) the error neighborhood is $\mathcal{O}(\epsilon^2 + \epsilon^4)$, which is guaranteed to be smaller than that of the SGD-GD-Based PP algorithm in (Li and Wai, 2024) when the distribution map is mildly sensitive ($\epsilon < 1$).

IFO complexity. We next derive the incremental first-order oracle (IFO) complexity (Reddi et al., 2016) of SPRINT —the number of per-sample gradient evaluations required to reach a $\mathcal{O}(\delta)$-SPS.

Corollary 5.5 (IFO Complexity of SPRINT)**.** *Consider the finite-population setting with n samples and $\alpha \in (0,1)$. With fixed parameters $\gamma_k = \gamma = \mathcal{O}(\frac{1}{n^\alpha}), \beta_k = \beta = \mathcal{O}(n^{\frac{\alpha}{2}}), m = \mathcal{O}(n^{\frac{\alpha}{2}})$, we have $\Gamma = \mathcal{O}(\frac{1}{n^\alpha})$, and the IFO complexity required to reach a $\mathcal{O}(\delta)$-SPS solution (beyond the error neighborhood) is $\mathcal{O}\left(\frac{(n^\alpha + n^{1+\frac{\alpha}{2}})\Delta_0}{\delta}\right)$.*

By Theorem 1 of Li and Wai (2024), the IFO complexity of SGD-GD is $\mathcal{O}\left(\frac{(\Delta_0 + \frac{L}{2}\sigma_0^2)^2}{\delta^2}\right)$. Comparing the two, SPRINT has a clear advantage when (i) the population size n is moderate and δ is small; or (ii) the variance parameter σ_0^2 is large or even intractable (e.g., heavy-tailed noise (Gurbuzbalaban et al., 2021)), in which case the SGD-GD bound can become arbitrarily large or even infinite. By contrast, SPRINT 's complexity is independent of σ_0. When the population is very large and δ is not too small, however, SPRINT can be slower than SGD-GD.

Extension to infinite populations. Our finite-sum analysis avoids the bounded-variance assumption by computing the full gradient at each snapshot. In the infinite-population setting this is no longer possible, and some form of tail control is needed to estimate the mean gradient. One option is to reinstate Assumption 3.4, or a milder bounded-$(1+\kappa)$-moment condition:

1. Use a minibatch as the gradient snapshot, which attains SVRG-type rates but requires a bounded-variance assumption.
2. Use a growing-minibatch control-variate scheme similar to Jothimurugesan et al. (2018), which keeps the gradient variance bounded without an explicit bounded-variance assumption.

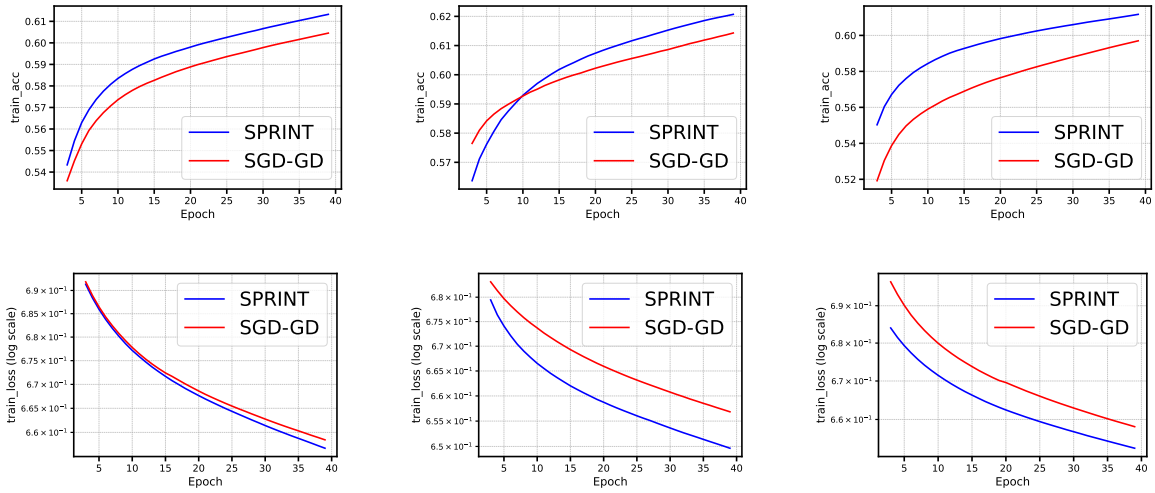
We emphasize that even in the finite-sum setting, SGD-GD still requires Assumption 3.4 because it samples a random component gradient at every iteration, so the per-iterate variance does not vanish. Thus the improvement of SPRINT over SGD-GD is robust.

6 Experiments

In this section, we conduct comprehensive experimental studies to verify the effectiveness of SPRINT in practice. We compare the convergence behavior of the training loss and training accuracy of SPRINT with SGD-GD (Li and Wai, 2024) on 3 real-world datasets with non-convex models and study the performative effects. The results of Credit dataset (Kaggle, 2012) and CIFAR-10 dataset (Krizhevsky and Hinton, 2009) are included in the main paper, while results of MNIST dataset (Deng, 2012) and additional experiments on different random seeds are relegated in App. F due to space limitation. We also show the magnitude of the squared gradient norm $\|\nabla \mathcal{J}(\widetilde{\boldsymbol{\theta}}^s; \widetilde{\boldsymbol{\theta}}^s)\|^2$ in App. F to validate Thm. 5.4.

Besides the credit and MNIST datasets, we are the first to provide results using CNN models on the CIFAR-10 (Krizhevsky and Hinton, 2009) dataset in the PP settings. As neural network architectures are widely used, our experiments go one step further for the practical applications of PP algorithms.

Credit dataset. We use the *Give me some credit* data (Kaggle, 2012) with 10 features $X \in \mathbb{R}^{10}$ to predict creditworthiness $Y \in \{0, 1\}$ (Perdomo et al., 2021; Jin et al., 2024). We preprocess the dataset by selecting a balanced subset of $5,000$ examples and then normalizing features similar to Perdomo et al. (2021). We then assume a credit scoring agency uses a two-layer MLP model to predict Y given the individuals' features X with cross entropy loss. To simulate performative effects, we assume individuals are strategic (Perdomo et al., 2021; Hardt et al., 2015), i.e., there is a subset of features $X_s \in X$ which individuals can change to

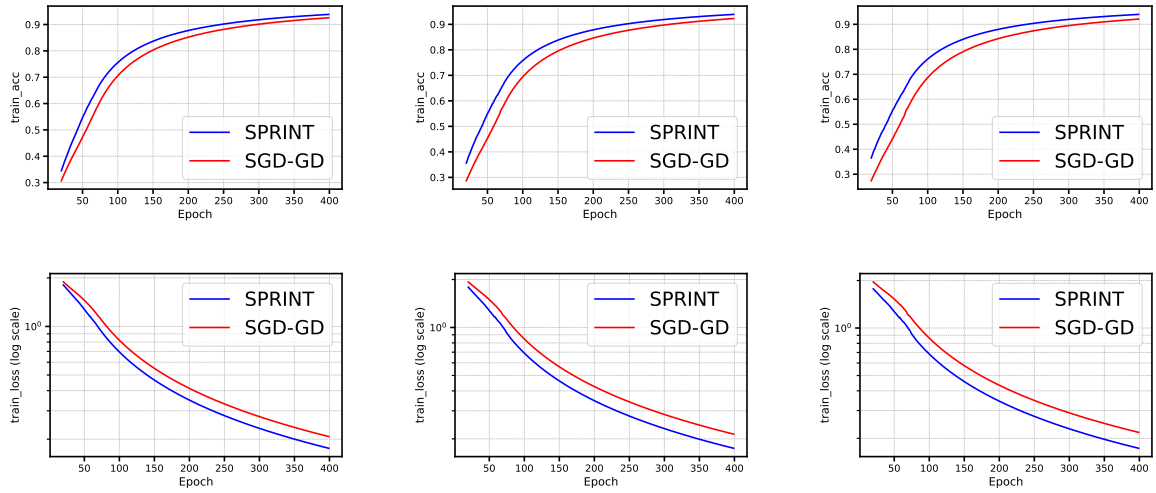


(a) $\alpha = 0.01$ (b) $\alpha = 0.2$ (c) $\alpha = 0.4$

Figure 1: The **Credit** dataset: training accuracy (first row) and training loss (second row) of SGD-GD and SPRINT under different α. A larger α implies more intense individual strategic behaviors.

X'_s based on the current model. Given the current model $f_{\boldsymbol{\theta}}$, individuals with feature x_s will best respond to the model in the direction of $\nabla_x f_{\boldsymbol{\theta}}(x_s)$ and change x_s to $x'_s = x_s + \alpha \nabla_x f_{\boldsymbol{\theta}}(x_s)$ to improve their scoring (Rosenfeld et al., 2020; Xie and Zhang, 2024).

In this setting, we train the MLP model using SPRINT and SGD-GD for 40 epochs and illustrate the accuracy and training loss in Fig. 1. We simulate different performative effects using $\alpha \in \{0.01, 0.2, 0.4\}$. In Fig. 1, we show that the SPRINT algorithm achieves lower training loss and converges faster compared with the SGD-GD in all 3 cases and the advantages are quite prominent when performative effects are larger (i.e., when $\alpha = 0.2$ and 0.4). To assess generalization, we additionally report out-of-sample evaluation accuracy on a held-out test set. In Appx. F, we shows that SPRINT achieves competitive or superior evaluation accuracy compared to SGD-GD across all α settings, indicating that faster convergence to SPS does not come at the cost of generalization.



(a) $\alpha = 20$ (b) $\alpha = 50$ (c) $\alpha = 80$

Figure 2: The **CIFAR-10** dataset: training accuracy (first row) and training loss (second row) of SGD-GD and SPRINT under different α. A larger α implies more intense retention dynamics.

CIFAR-10 dataset. Furthermore, we use the **CIFAR-10 dataset (Krizhevsky and Hinton, 2009)** to predict 10 possible class labels $Y \in \{0, 1, ..., 9\}$. Each label corresponds to a common and well-separated category such as *dog* and *truck*. The dataset contains $50,000$ images in total and each class includes $5,000$ samples. We design a CNN model with two convolutional layers to train the algorithms with a 0.05 learning rate and cross-entropy loss. We simulate performative effects motivated by retention dynamics (Hashimoto et al., 2018; Jin et al., 2024; Zhang et al., 2019) where the class distribution will change based on the performance of the current model. The fraction in class c at iteration k of $p_c^{k+1} = \frac{e^{-\alpha \ell_c^k}}{\sum_{c'} e^{-\alpha \ell_{c'}^k}}$, where ℓ_c^k is the loss expectation of class c in iteration k. This means that the image class with larger loss now will be less likely to appear in the next iteration (Zhang et al., 2019). We set α at $20, 50, 80$ and train 400 epochs in total. The experimental results are shown in Fig. 2. Consistent with the results observed on the Credit dataset, SPRINT also exhibits a faster convergence on the CIFAR-10 dataset. We further note that the CIFAR-10 experiments employ CNN architectures for which Assumption 3.3 may not hold globally. The stable convergence observed empirically suggests that SPRINT 's performance is robust to mild violations of this assumptions in practice.

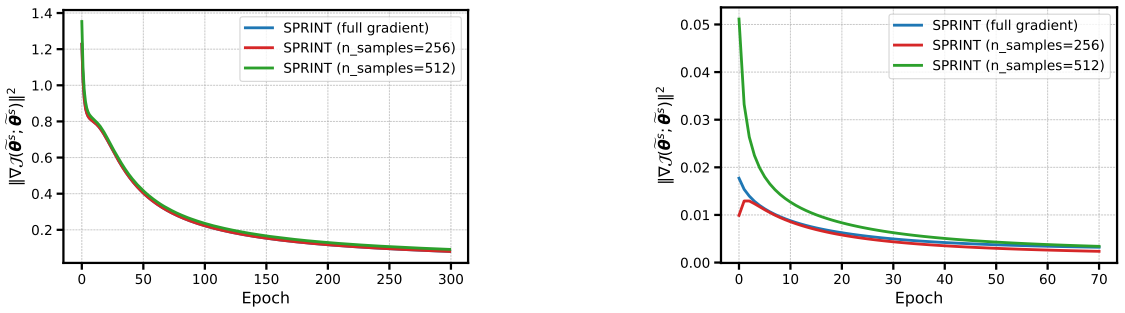
Mini-batch gradient snapshot experiments. While the theoretical guarantee of SPRINT requires a full gradient snapshot computed over the entire population, this requirement may be impratical in large-population settings where iterating over all n samples at the beginning of each epoch incurs prohibitive computational cost. A natural remedy is to approximate the full gradient snapshot with a mini-batch of size $B \ll n$. Although Theorem 5.4 does not apply to this setting, it is important to assess whether SPRINT degrades gracefully under such approximations in practice.

To this end, we investigate the empirical behavior of SPRINT when the snapshot gradient $\nabla \mathcal{J}(\widetilde{\boldsymbol{\theta}}^s; \widetilde{\boldsymbol{\theta}}^s)$ is approximated by mini-batches of size $B \in \{256, 512\}$, and compare against the full-gradient implementation. We report the squared gradient norm $\|\nabla \mathcal{J}(\widetilde{\boldsymbol{\theta}}^s; \widetilde{\boldsymbol{\theta}}^s)\|^2$ across epochs on both the Credit and MNIST datasest in Figures 3a and 3b.

Several observations emerge from the results. First, across both datasets, the convergence curves under $B = 256$ and $B = 512$ remain close to that of the full-gradient variant throughout training, suggesting that the variance introduced by the mini-batch snapshot is well-controlled in practice. Second, the gap between different batch sizes is small and does not widen appreciably as training progresses, indicating that SPRINT is not overly sensitive to the fidelity of the gradient snapshot. Third, the ordering of curves is consistent with intuition: a larger batch size yields a curve closer to the full-gradient baseline, as the snapshot estimate is more accurate. These results suggest that SPRINT is practically deployable even in large-population settings where a full gradient snapshot is too costly, and that a modest batch size such as $B = 512$ may suffice to retain most of the convergence benefits.

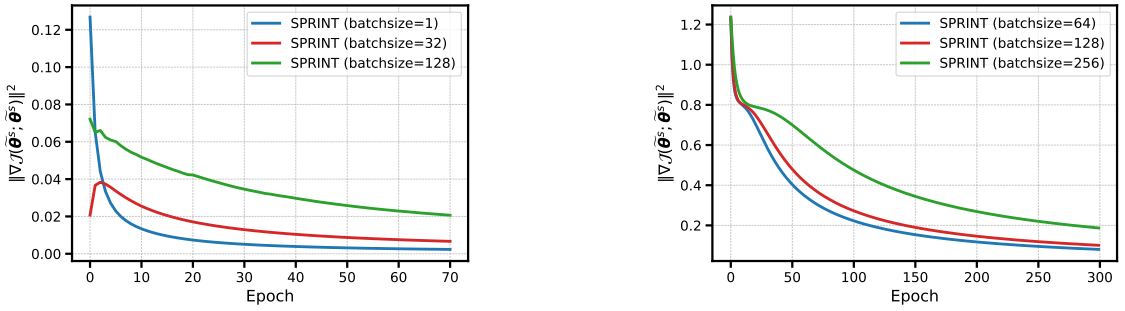
Empirical validation of variance-independent error neighborhood. The key observation is that the batch size B used during the inner iterations of SPRINT directly controls the stochastic gradient variance: a larger B yields a smaller per-iterate variance σ_0^2, while a smaller B corresponds to noisier gradient estimates. If the converged error neighborhood were variance-dependent, one would expect the gradient norm at convergence $\|\nabla \mathcal{J}(\widetilde{\boldsymbol{\theta}}^s; \widetilde{\boldsymbol{\theta}}^s)\|^2$ to decrease monotonically as B increases. Conversely, if the error neighborhood is truly variance-independent as Theorem 5.4 predicts, the converged gradient norm should remain approximately constant across batch sizes.

We conduct this experiment on both the Credit and MNIST datasets by varying $B \in \{1, 32, 128\}$ for Credit dataset and $B \in \{64, 128, 256\}$ for MNIST dataset, while keeping the full gradient snapshot fixed. The results are shown in Figures 4a and Figures 4b. Across both datasets and all batch sizes tested, the converged gradient norms of SPRINT cluster tightly together and do not exhibit a clear decreasing trend as B grows. This stands in contrast to what one would expect from a variance-dependent algorithm, and is consistent with the theoretical prediction of Theorem 5.4. We further note that the batch size primarily affects the transient convergence speed but the final error level remains stable, providing strong empirical support for the variance-independent nature of SPRINT 's error neighborhood.



(a) Credit dataset (b) MNIST dataset

Figure 3: Mini-batch gradient snapshot experiments on the **Credit** and **MNIST** datasets. We plot the squared gradient norm $\|\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)\|^2$ over epochs for SPRINT with a full gradient snapshot and with mini-batch snapshot sizes $B \in \{256, 512\}$. A larger batch size yields a snapshot estimate closer to the true full gradient and thus a convergence curve nearer to the full-gradient baseline. The results demonstrate that SPRINT degrades only mildly under moderate snapshot approximation, suggesting practical deployability in large-population settings.



(a) Credit dataset (b) MNIST dataset

Figure 4: Empirical validation of variance-independent error neighborhood on the Credit dataset (left) and MNIST dataset (right). We vary the batch size $B \in \{1, 32, 128\}$ (Credit) and $B \in \{64, 128, 256\}$(MNIST) to control the stochastic gradient variance σ_0^2. A larger batch size corresponds to smaller variance. If the error neighborhood were variance-dependent (as in SGD-GD), the converged gradient norm $\nabla\|\mathcal{J}(\tilde{\theta}^s, \tilde{\theta}^s)\|^2$ would decrease as batch size increases. The results show that the converged gradient norms of SPRINT remain approximately constant across different batch sizes, providing empirical evidence consistent with the variance-independent error neighborhood established in Theorem 5.4.

7 Conclusion and future work

This paper focused on improving the convergence of stochastic optimization methods in non-convex PP settings by adapting a variance reduction method. We proposed SPRINT as a new algorithm to achieve better convergence results without the need for the bounded variance assumption on stochastic gradient descent estimates. We also presented extensive experiments with different non-convex neural network models to compare SPRINT with SGD-GD, which demonstrated the superior performance of our algorithm SPRINT. However, SPRINT only converges to a non-vanishing error neighborhood. Future work may include developing algorithms to converge to performative optimal solutions in the non-convex PP settings, or adapting other more advanced variance reduction methods (SARAH or SPIDER) to the PP setting and trying to completely eliminate the non-vanishing error neighborhood.

References

Arpit Agarwal and Shivani Agarwal. On consistent surrogate risk minimization and property elicitation. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 4–22, Paris, France, 03–06 Jul 2015. PMLR. URL https://proceedings.mlr.press/v40/Agarwal15.html.

Ioannis Anagnostides, Rohan Chauhan, Ioannis Panageas, Tuomas Sandholm, and Jingming Yan. On the computational complexity of performative prediction, 2026. URL https://arxiv.org/abs/2601.20180.

Alan W Beggs. On the convergence of reinforcement learning. *Journal of economic theory*, 122(1):1–36, 2005.

Gavin Brown, Shlomi Hod, and Iden Kalemaj. Performative prediction in a stateful world. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 6045–6061. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/brown22a.html.

Jianfeng Chi, Jian Shen, Xinyi Dai, Weinan Zhang, Yuan Tian, and Han Zhao. Towards return parity in markov decision processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1161–1178. PMLR, 2022.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Dmitriy Drusvyatskiy and Lin Xiao. Stochastic optimization with decision-dependent distributions. *Mathematics of Operations Research*, 48(2):954–998, 2023.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in neural information processing systems*, 31, 2018.

Mert Gurbuzbalaban, Umut Simsekli, and Lingjiong Zhu. The heavy-tail phenomenon in sgd. In *International Conference on Machine Learning*, pages 3964–3975. PMLR, 2021.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic Classification, November 2015. URL http://arxiv.org/abs/1506.06980. Number: arXiv:1506.06980 arXiv:1506.06980 [cs].

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

Zachary Izzo, Lexing Ying, and James Zou. How to learn when data reacts to your model: Performative gradient descent. *CoRR*, 2021. arXiv: 2102.07698.

Kun Jin, Tian Xie, Yang Liu, and Xueru Zhang. Addressing polarization and unfairness in performative prediction, 2024. URL https://arxiv.org/abs/2406.16756.

Ellango Jothimurugesan, Ashraf Tahmasbi, Phillip Gibbons, and Srikanta Tirthapura. Variance-reduced stochastic gradient descent on streaming data. *Advances in neural information processing systems*, 31, 2018.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Kaggle. Give me some credit. https://www.kaggle.com/c/GiveMeSomeCredit/data, 2012.

Michael P Kim and Juan C Perdomo. Making decisions under outcome performativity. *arXiv preprint arXiv:2210.01745*, 2022.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Qiang Li and Hoi-To Wai. State dependent performative prediction with stochastic approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 3164–3186. PMLR, 2022.

Qiang Li and Hoi-To Wai. Stochastic optimization schemes for performative prediction with nonconvex loss. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Qiang Li, Michal Yemini, and Hoi-To Wai. Clipped sgd algorithms for performative prediction: tight bounds for stochastic bias and remedies. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Haitong Liu, Qiang Li, and Hoi-To Wai. Two-timescale derivative free optimization for performative prediction with markovian data. *arXiv preprint arXiv:2310.05792*, 2023.

Celestine Mendler-Dünner, Frances Ding, and Yixin Wang. Anticipating performativity by predicting from predictions. *Advances in Neural Information Processing Systems*, 35:31171–31185, 2022.

Celestine Mendler-Dünner, Juan Perdomo, Tijana Zrnic, and Moritz Hardt. Stochastic optimization for performative prediction. In *Advances in neural information processing systems*, pages 4929–4939. Curran Associates, Inc., 2020.

John P Miller, Juan C Perdomo, and Tijana Zrnic. Outside the echo chamber: Optimizing the performative risk. In *International Conference on Machine Learning*, pages 7710–7720. PMLR, 2021.

Mehrnaz Mofakhami, Ioannis Mitliagkas, and Gauthier Gidel. Performative prediction with neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 11079–11093. PMLR, 2023.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.

Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative Prediction. *arXiv:2002.06673 [cs, stat]*, February 2021. URL `http://arxiv.org/abs/2002.06673`. arXiv: 2002.06673.

Mitas Ray, Lillian J Ratliff, Dmitriy Drusvyatskiy, and Maryam Fazel. Decision-dependent risk minimization in geometrically decaying dynamic environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8081–8088, 2022.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323. PMLR, 2016.

Nir Rosenfeld, Anna Hilgard, Sai Srivatsa Ravindranath, and David C Parkes. From predictions to decisions: Using lookahead regularization. *Advances in Neural Information Processing Systems*, 33:4115–4126, 2020.

Nicolas Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. *Advances in neural information processing systems*, 25, 2012.

Emanuel Todorov et al. Optimal control theory. *Bayesian brain: probabilistic approaches to neural coding*, pages 268–298, 2006.

Nikita Tsoy, Ivan Kirev, Negin Rahimi, and Nikola Konstantinov. On the impact of performative risk minimization for binary random variables. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282): 1–36, 2022.

Tian Xie and Xueru Zhang. Non-linear welfare-aware strategic learning, 2024.

Tian Xie, Xuwei Tan, and Xueru Zhang. Algorithmic decision-making under agents with persistent improvement, 2024.

Rui Yuan, Robert M Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In *International Conference on Artificial Intelligence and Statistics*, pages 3332–3380. PMLR, 2022.

Xueru Zhang, Mohammad Mahdi Khalili, Cem Tekin, and Mingyan Liu. *Group retention when using machine learning in sequential decision making: the interplay between user dynamics and fairness*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Xue Zheng, Tian Xie, Xuwei Tan, Aylin Yener, Xueru Zhang, Ali Payani, and Myungjin Lee. Profl: Performative robust optimal federated learning. *arXiv preprint arXiv:2410.18075*, 2024.

Zihan Zhu, Ethan Fang, and Zhuoran Yang. Online performative gradient descent for learning nash equilibria in decision-dependent games. *Advances in Neural Information Processing Systems*, 36:47902–47913, 2023.

A The Use of Large Language Models (LLMs)

We only use LLMs for polishing the writing without using them to generate any single sentence solely on their own.

B Proof of Lemma 5.1

Lemma. Let β_k be some positive constant, and c_k, c_{k+1} are some constants defined in intermediate functions R_k^{s+1}, R_{k+1}^{s+1} that satisfy the following

$$c_k - \frac{1}{2}\gamma_k = c_{k+1}(1 + \gamma_k\beta_k + 2L_0\epsilon\gamma_k) + (2L^2 + L_0^2\epsilon^2)(L\gamma_k^2 + 2c_{k+1}\gamma_k^2) + \frac{\beta_k}{2}L_0\epsilon\gamma_k$$

Then, we have

$$R_{k+1}^{s+1}(\boldsymbol{\theta}_k^{s+1}) \leq R_k^{s+1}(\boldsymbol{\theta}_k^{s+1}) - \Gamma_k \cdot \mathbb{E}[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2] - \frac{1}{2}\gamma_k\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2$$

where $\Gamma_k = \left(\gamma_k - \frac{c_{k+1}\gamma_k + \frac{1}{2}L_0\epsilon\gamma_k}{\beta_k} - 2L\gamma_k^2 - 4c_{k+1}\gamma_k^2\right)$

Proof. This proof mainly relies on Assumptions 3.1, 3.3, and ϵ-sensitivity.

We first prove another auxiliary Lemma which is very similar to Lemma 5.2.

Lemma B.1. *Under Assumption 3.1 and the ϵ-sensitivity condition in Def. 3.2, for any $\boldsymbol{\theta}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$, we have:*

$$\big\|\nabla J(\boldsymbol{\theta};\boldsymbol{\theta}_1) - \nabla J(\boldsymbol{\theta};\boldsymbol{\theta}_2)\big\| \leq L_0\,\epsilon\,\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|. \tag{10}$$

Proof. Let $\Pi(\mathcal{D}(\boldsymbol{\theta}_1), D(\boldsymbol{\theta}_2))$ denote the set of all couplings between $\mathcal{D}(\boldsymbol{\theta}_1)$ and $\mathcal{D}(\boldsymbol{\theta}_2)$. For any $\pi \in \Pi(\mathcal{D}(\boldsymbol{\theta}_1), \mathcal{D}(\boldsymbol{\theta}_2))$, let $(Z_1, Z_2) \sim \pi$. Then

$$\big\|\nabla\mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}_1) - \nabla\mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}_2)\big\| = \Big\|\mathbb{E}\big[\nabla\ell(Z_1;\boldsymbol{\theta}) - \nabla\ell(Z_2;\boldsymbol{\theta})\big]\Big\| \leq \mathbb{E}\Big[\big\|\nabla\ell(Z_1;\boldsymbol{\theta}) - \nabla\ell(Z_2;\boldsymbol{\theta})\big\|\Big] \leq L_0\,\mathbb{E}\big[\|Z_1 - Z_2\|\big],$$

where the last inequality follows from Assumption 3.1, i.e., $\|\nabla\ell(z;\boldsymbol{\theta}) - \nabla\ell(z';\boldsymbol{\theta})\| \leq L_0\|z - z'\|$ for all z, z'. Thus,

$$\big\|\nabla\mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}_1) - \nabla\mathcal{J}(\boldsymbol{\theta};\boldsymbol{\theta}_2)\big\| \leq L_0\,W_1\big(\mathcal{D}(\boldsymbol{\theta}_1), \mathcal{D}(\boldsymbol{\theta}_2)\big).$$

Finally, applying the ϵ-sensitivity condition $W_1(\mathcal{D}(\boldsymbol{\theta}_1), \mathcal{D}(\boldsymbol{\theta}_2)) \leq \epsilon\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|$ completes the proof. □

Now we begin to prove Lemma 5.1. Since $\ell(z;\boldsymbol{\theta})$ is L-smooth in $\boldsymbol{\theta}$ and $v_k^{s+1} = \frac{1}{\gamma_k}\big(\boldsymbol{\theta}_k^{s+1} - \boldsymbol{\theta}_{k+1}^{s+1}\big)$, we can apply descent lemma to get the following result:

$$\begin{aligned}\mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_k^{s+1})] &\leq \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1}) + \frac{L}{2}\|\boldsymbol{\theta}_{k+1}^{s+1} - \boldsymbol{\theta}_k^{s+1}\|^2 + \nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})^T(\boldsymbol{\theta}_{k+1}^{s+1} - \boldsymbol{\theta}_k^{s+1})] \\ &= \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})] + \frac{L\gamma_k^2}{2}\|v_k^{s+1}\|^2 - \gamma_k\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})^T v_k^{s+1}\end{aligned} \tag{11}$$

By definition of v_k^{s+1}, we have:

$$\begin{aligned}\mathbb{E}[v_k^{s+1}] &= \mathbb{E}[\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1}) - \nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s) + \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)] \\ &= \mathbb{E}[\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1}) - \nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s) + \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})] - \mathbb{E}[\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1}) - \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)] \\ &= \mathbb{E}[\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})] - \mathbb{E}[\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1}) - \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)]\end{aligned} \tag{12}$$

The last equality holds since $\mathbb{E}[\nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s)] = \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s,\boldsymbol{\theta}_k^{s+1})$. Then applying Lemma B.1 on the second term above to derive the following:

$$\begin{aligned}\gamma_k\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})^T v_k^{s+1} &= \gamma_k\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})^T\big(\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1}) - \mathbb{E}[\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1}) - \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)]\big)\\ &\geq \gamma_k\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2 - L_0\epsilon\gamma_k\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|\|\widetilde{\boldsymbol{\theta}}^s - \boldsymbol{\theta}_k^{s+1}\|\end{aligned}$$

Next, applying Young's Inequality, we can get:

$$\begin{aligned}\text{Eqn. 11} \leq \mathbb{E}[&\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_k^{s+1}) - \gamma_k\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2 + L_0\epsilon\gamma_k(\frac{1}{2\beta_k}\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\\ &+ \frac{\beta_k}{2}\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2) + \frac{L\gamma_k^2}{2}\|v_k^{s+1}\|^2]\end{aligned} \tag{13}$$

Next, bound $\mathbb{E}[\|\boldsymbol{\theta}_{k+1}^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2]$ by adding and subtracting as follows:

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{\theta}_{k+1}^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2] &= \mathbb{E}[\|\boldsymbol{\theta}_{k+1}^{s+1} - \boldsymbol{\theta}_k^{s+1} + \boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2]\\ &= \mathbb{E}[\gamma_k^2\|v_k^{s+1}\|^2 + \|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2] + 2\gamma_k\mathbb{E}[\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|\|v_k^{s+1}\|]\\ &\leq \mathbb{E}[\gamma_k^2\|v_k^{s+1}\|^2 + \|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2] + 2\gamma_k\mathbb{E}[\frac{1}{2\beta_k}\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\\ &\quad + \frac{\beta_k}{2}\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2] + 2L_0\epsilon\gamma_k\mathbb{E}\|\widetilde{\boldsymbol{\theta}}^s - \boldsymbol{\theta}_k^{s+1}\|^2\end{aligned} \tag{14}$$

Similary, $\mathbb{E}[\|v_k^{s+1}\|^2]$ can also be bounded:

$$\begin{aligned}\mathbb{E}[\|v_k^{s+1}\|^2] &= \mathbb{E}\left[\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1}) - \nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s) + \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)\right]^2\\ &\leq 2\mathbb{E}\left[\|\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1}) - \nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s) + \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})\|^2\right] + 2\mathbb{E}\left[\|\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1}) - \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)\|^2\right]\end{aligned} \tag{15}$$

The above inequality is obtained by adding and subtracting $\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})$ and then applying Young's Inequality. For the first term, it is worth noting that the term already gets rid of the performative effects, i.e., the distribution is fixed as $\mathcal{D}(\boldsymbol{\theta}_k^{s+1})$. Then bounding this term is equivalent to bound the norm of update in Reddi et al. (2016). Specifically, let $\delta_k^{s+1} = \nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1}) - \nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s)$, we know $\mathbb{E}[\delta_k^{s+1}] = \mathbb{E}[\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1})] - \mathbb{E}[\nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s)] = \nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1}) - \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})$. Thus, we have:

$$\begin{aligned}&\mathbb{E}\left[\|\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1}) - \nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s) + \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})\|^2\right]\\ &= \mathbb{E}[\|\delta_k^{s+1} + \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})\|^2]\\ &= \mathbb{E}[\|\delta_k^{s+1} + \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1}) - \nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1}) + \nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2]\\ &= \mathbb{E}[\|\delta_k^{s+1} - \mathbb{E}[\delta_k^{s+1}] + \nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2]\\ &\leq 2\mathbb{E}[\|\delta_k^{s+1} - \mathbb{E}[\delta_k^{s+1}]\|^2] + 2\mathbb{E}[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2]\\ &\leq 2\mathbb{E}[\|\delta_k^{s+1}\|^2] + 2\mathbb{E}[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2]\\ &\leq 2L^2\mathbb{E}[\|\widetilde{\boldsymbol{\theta}}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2] + 2\mathbb{E}[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2]\end{aligned} \tag{16}$$

By Lemma B.1, we know $\|\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1}) - \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)\| \leq L_0\epsilon\|\widetilde{\boldsymbol{\theta}}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|$ always holds. So we have:

$$\mathbb{E}\left[\|\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1}) - \nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)\|^2\right] \leq L_0^2\epsilon^2\mathbb{E}\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2 \tag{17}$$

Sum them together, we have:

$$\mathbb{E}[\|v_k^{s+1}\|^2] \leq 4\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2 + (4L^2 + 2L_0^2\epsilon^2)\mathbb{E}\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2 \tag{18}$$

Now consider R_{k+1}^{s+1} and plug Eqn. 11 and Eqn. 14 into its formula, we have:

$$\begin{aligned}
R_{k+1}^{s+1}(\boldsymbol{\theta}_k^{s+1}) &= \mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_k^{s+1}) + c_{k+1}\|\widetilde{\boldsymbol{\theta}}_{k+1}^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2\right] \\
&\leq \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1}) - \gamma_k\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1},\boldsymbol{\theta}_k^{s+1})\|^2 + L_0\epsilon\gamma_k(\frac{1}{2\beta_k}\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2 + \frac{\beta_k}{2}\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2) \\
&\quad + \frac{L\gamma_k^2}{2}\|v_k^{s+1}\|^2] + \mathbb{E}[c_{k+1}\gamma_k^2\|v_k^{s+1}\|^2 + c_{k+1}\|\boldsymbol{\theta}_{k+1}^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2 + 2c_{k+1}\gamma_k\mathbb{E}(\frac{1}{2\beta_k}\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2 \\
&\quad + \frac{\beta_k}{2}\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2)] + 2c_{k+1}L_0\epsilon\gamma_k\mathbb{E}\left\|\widetilde{\boldsymbol{\theta}}^s - \boldsymbol{\theta}_k^{s+1}\right\|^2 \\
&= \mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\right] + \left(\frac{L\gamma_k^2}{2} + c_{k+1}\gamma_k^2\right)\mathbb{E}\left[\|v_k^{s+1}\|^2\right] \\
&\quad - \left(\gamma_k - \frac{c_{k+1}\gamma_k + \frac{1}{2}L_0\epsilon\gamma_k}{\beta_k}\right)\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right] \\
&\quad + \left(c_{k+1}(1 + \gamma_k\beta_k + 2L_0\epsilon\gamma_k) + \frac{\beta_k L_0\epsilon\gamma_k}{2}\right)\mathbb{E}\left[\|\widetilde{\boldsymbol{\theta}}^s - \boldsymbol{\theta}_k^{s+1}\|^2\right]
\end{aligned}$$

Regarding the $\|v_k^{s+1}\|^2$, we plug Eqn. 18 in Eqn. 19 to get the expression we need:

$$\begin{aligned}
R_{k+1}^{s+1}(\boldsymbol{\theta}_k^{s+1}) &\leq \mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\right] - \Gamma_k\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right] + (c_k - \frac{1}{2}\gamma_k)\mathbb{E}\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2 \\
&= R_k^{s+1} - \Gamma_k\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right] - \frac{1}{2}\gamma_k\mathbb{E}\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2
\end{aligned}$$

where

$$\begin{aligned}
c_k - \frac{1}{2}\gamma_k &= c_{k+1}(1 + \gamma_k\beta_k + 2L_0\epsilon\gamma_k) + (2L^2 + L_0^2\epsilon^2)(L\gamma_k^2 + 2c_{k+1}\gamma_k^2) + \frac{\beta_k}{2}L_0\epsilon\gamma_k \\
\Gamma_k &= \left(\gamma_k - \frac{c_{k+1}\gamma_k + \frac{1}{2}L_0\epsilon\gamma_k}{\beta_k} - 2L\gamma_k^2 - 4c_{k+1}\gamma_k^2\right)
\end{aligned}$$

□

C Proof of Lemma 5.3

Lemma. With $c_k, c_{k+1}, \beta_k, \Gamma_k$ same as Lemma 5.1, denote

$$\begin{aligned}
\widetilde{R}_k^{s+1} &\triangleq \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1}) + c_k\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2] \\
\widetilde{R}_{k+1}^{s+1} &\triangleq \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_{k+1}^{s+1}) + c_{k+1}\|\boldsymbol{\theta}_{k+1}^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2]
\end{aligned}$$

Then we have:

$$\widetilde{R}_{k+1}^{s+1} \leq \widetilde{R}_k^{s+1} + (L_0^2L_1^2\epsilon^4 + 2L_1^2\epsilon^2L^2 + 2L_1^2\epsilon^2)\gamma_k - (\Gamma_k - \frac{\gamma_k}{4}) \cdot \mathbb{E}[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2] \tag{19}$$

Proof. This proof mainly relies on Assumptions 3.1, 3.3, and ϵ-sensitivity.

According to Lemma 5.1, we have:

$$\begin{aligned}
\mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_k^{s+1}) + c_{k+1}\|\boldsymbol{\theta}_{k+1}^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2] &\leq \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})] + c_k\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2 \\
&\quad - \Gamma_k\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2 - \frac{\gamma_k}{2}\|\boldsymbol{\theta}_k^{s+1} - \widetilde{\boldsymbol{\theta}}^s\|^2]
\end{aligned} \tag{20}$$

By aggregating terms, we will have:

$$\begin{aligned} & c_{k+1}\mathbb{E}\|\boldsymbol{\theta}_{k+1}^{s+1}-\widetilde{\boldsymbol{\theta}}^{s}\|^2-\left(c_k-\frac{\gamma_k}{2}\right)\mathbb{E}\|\boldsymbol{\theta}_{k}^{s+1}-\widetilde{\boldsymbol{\theta}}^{s}\|^2+\Gamma_k\mathbb{E}\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2 \\ \leq\ & \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})-\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_k^{s+1})] \end{aligned} \tag{21}$$

We can further bound the above expression by:

$$\begin{aligned} \mathbb{E}[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})-\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_k^{s+1})] &= \mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})-\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_{k+1}^{s+1})\right] \\ &\quad+\mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_{k+1}^{s+1})-\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_k^{s+1})\right] \\ &\leq \mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})-\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_{k+1}^{s+1})\right] \\ &\quad+L_1\epsilon\mathbb{E}\|\boldsymbol{\theta}_{k+1}^{s+1}-\boldsymbol{\theta}_k^{s+1}\| \end{aligned} \tag{22}$$

Regarding $\mathbb{E}\|\boldsymbol{\theta}_{k+1}^{s+1}-\boldsymbol{\theta}_k^{s+1}\|=\gamma_k\mathbb{E}\|v_k^{s+1}\|$, we can bound it according to Eqn. 12 and Eqn. 18:

$$\begin{aligned} \mathbb{E}[\|v_k^{s+1}\|] &= \mathbb{E}[\|\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1})-\nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s)+\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)\|] \\ &\leq \mathbb{E}[\|\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1})-\nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s)+\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})\|] \\ &\quad+\mathbb{E}[\|\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})-\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)\|] \end{aligned} \tag{23}$$

Let $\nabla\ell(z_{i_k};\boldsymbol{\theta}_k^{s+1})-\nabla\ell(z_{i_k};\widetilde{\boldsymbol{\theta}}^s)+\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})$ be A and $\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\boldsymbol{\theta}_k^{s+1})-\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s;\widetilde{\boldsymbol{\theta}}^s)$ be B. According to Eqn. 16, we have:

$$\mathbb{E}[\|A\|^2]\leq 2\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right]+2L^2\mathbb{E}\|\boldsymbol{\theta}_k^{s+1}-\widetilde{\boldsymbol{\theta}}^s\|^2$$

which means

$$\mathbb{E}[\|A\|]\leq\sqrt{\mathbb{E}[\|A\|^2]}\leq\sqrt{2\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right]+2L^2\mathbb{E}\|\boldsymbol{\theta}_k^{s+1}-\widetilde{\boldsymbol{\theta}}^s\|^2}$$

Applying Young's Inequality on rhs, for any positive λ we have:

$$\mathbb{E}\|A\|\leq\frac{1}{\lambda}\left(\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right]+L^2\mathbb{E}\left\|\boldsymbol{\theta}_k^{s+1}-\widetilde{\boldsymbol{\theta}}^s\right\|^2\right)+\frac{\lambda}{2} \tag{24}$$

Let $\lambda_0=4L_1\epsilon(L^2+1)$, we have:

$$\begin{aligned} \mathbb{E}\|A\| &\leq \frac{1}{\lambda_0}\left(\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right]+L^2\mathbb{E}\left\|\boldsymbol{\theta}_k^{s+1}-\widetilde{\boldsymbol{\theta}}^s\right\|^2\right)+\frac{\lambda_0}{2} \\ &\leq \frac{1}{4L_1\epsilon}\mathbb{E}\left[\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2\right]+\frac{1}{4L_1\epsilon}\mathbb{E}\left\|\boldsymbol{\theta}_k^{s+1}-\widetilde{\boldsymbol{\theta}}^s\right\|^2+\frac{\lambda_0}{2} \end{aligned} \tag{25}$$

According to the point-wise Lipschitzness in Lemma B.1, we have:

$$\mathbb{E}[\|B\|]\leq\sqrt{\mathbb{E}[\|B\|^2]}\leq\sqrt{L_0^2\epsilon^2\mathbb{E}\left\|\boldsymbol{\theta}_k^{s+1}-\boldsymbol{\theta}^s\right\|^2}\leq\frac{\mathbb{E}\left\|\boldsymbol{\theta}_k^{s+1}-\widetilde{\boldsymbol{\theta}}^s\right\|^2}{4L_1\epsilon}+L_0^2L_1\epsilon^3 \tag{26}$$

Thus, we have

$$\begin{aligned} \mathbb{E}[\|v_k^{s+1}\|] &\leq \mathbb{E}\|A\|+\mathbb{E}\|B\| \\ &\leq \frac{\mathbb{E}\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2}{4L_1\epsilon}+(\frac{1}{4L_1\epsilon}+\frac{1}{4L_1\epsilon})\,\mathbb{E}\left\|\boldsymbol{\theta}_k^{s+1}-\widetilde{\boldsymbol{\theta}}^s\right\|^2+\frac{\lambda_0}{2}+L_0^2L_1\epsilon^3 \\ &= \frac{\mathbb{E}\|\nabla\mathcal{J}(\boldsymbol{\theta}_k^{s+1};\boldsymbol{\theta}_k^{s+1})\|^2}{4L_1\epsilon}+\frac{1}{2L_1\epsilon}\mathbb{E}\left\|\boldsymbol{\theta}_k^{s+1}-\widetilde{\boldsymbol{\theta}}^s\right\|^2+(\frac{\lambda_0}{2}+L_0^2L_1\epsilon^3) \end{aligned} \tag{27}$$

Finally, we plug Eqn. 27 into Eqn. 22, and then plug Eqn. 22 into Eqn. 21. This leads to the inequality relationship between $\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_{k+1}^{s+1})$ and $\mathcal{J}(\boldsymbol{\theta}_{k}^{s+1};\boldsymbol{\theta}_{k}^{s+1})$.

$$\begin{aligned}\mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_{k+1}^{s+1};\boldsymbol{\theta}_{k+1}^{s+1})+c_{k+1}\|\boldsymbol{\theta}_{k+1}^{s+1}-\widetilde{\boldsymbol{\theta}}^{s}\|^2\right] &\leq \mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}_{k}^{s+1};\boldsymbol{\theta}_{k}^{s+1})+c_{k}\|\boldsymbol{\theta}_{k}^{s+1}-\widetilde{\boldsymbol{\theta}}^{s}\|^2\right]\\ &-(\Gamma_k-\frac{\gamma_k}{4})\cdot\mathbb{E}[\|\nabla\mathcal{J}(\boldsymbol{\theta}_{k}^{s+1};\boldsymbol{\theta}_{k}^{s+1})\|^2]+(L_0^2L_1^2\epsilon^4+2L_1^2\epsilon^2L^2+2L_1^2\epsilon^2)\gamma_k\end{aligned}$$

which is exactly what we want. □

D Proof of Thm. 5.4

Theorem. Let the total rounds be T and the epochs be $S=\lceil T/m\rceil$. For the last epoch, $c_m=0$. Define Δ_0 and Δ_1 as follows:

$$\Delta_0 \triangleq \mathcal{J}(\boldsymbol{\theta}_0^0;\boldsymbol{\theta}_0^0)-\ell_*; \qquad \Delta_1 \triangleq \frac{(L_0^2L_1^2\epsilon^4+2L_1^2\epsilon^2L^2+2L_1^2\epsilon^2)\sum_{s=0}^{S}\sum_{k=0}^{m}\gamma_k}{T\Gamma}$$

where $\Gamma>0$ is guaranteed to exist as a constant and is the lower bound of $\Gamma_k-\frac{\gamma_k}{4}$. Then we have:

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{k=0}^{m-1}\mathbb{E}\|\nabla\mathcal{J}(\boldsymbol{\theta}_{k}^{s+1};\boldsymbol{\theta}_{k}^{s+1})\|^2 \leq \frac{\Delta_0}{T\Gamma}+\Delta_1 \tag{28}$$

Thus, Alg. 1 can achieve $\mathcal{O}(\frac{1}{T})$ convergence rate and the error neighborhood Δ_1 is $\mathcal{O}(\epsilon^2+\epsilon^4)$.

Proof. Note the γ_t is just an abbreviation of the learning rate at global round s and local round k. Using Lemma 5.3 (Eqn. 8) and telescoping:

$$\sum_{s=0}^{S-1}\sum_{k=0}^{m-1}(\Gamma_k-\frac{\gamma_k}{4})\cdot\mathbb{E}[\|\nabla\mathcal{J}(\boldsymbol{\theta}_{k}^{s+1};\boldsymbol{\theta}_{k}^{s+1})\|^2] \leq (\widetilde{R}_0^1-\widetilde{R}_m^S)+(L_0^2L_1^2\epsilon^4+2L_1^2\epsilon^2L^2+2L_1^2\epsilon^2)\sum_{t=0}^{T-1}\gamma_t \tag{29}$$

Thus,

$$\frac{1}{T}\cdot\sum_{s=0}^{S-1}\sum_{k=0}^{m-1}\mathbb{E}[\|\nabla\mathcal{J}(\boldsymbol{\theta}_{k}^{s+1};\boldsymbol{\theta}_{k}^{s+1})\|^2] \leq \frac{\Delta_0}{T\Gamma}+\Delta_1$$

There is one remaining step: prove there exists values of γ_k,β_k to let $\Gamma>0$. We first begin with a lemma:

Lemma D.1. *For any $\eta>0$ and β_k is some constant β, there exists a sequence of learning rate $\{\gamma_k\}_{k=1}^m$ to let $c_k\leq\eta L_0\epsilon$ hold for any k.*

Proof. We can start from $c_m=0$ and are given the relationship between c_k and c_{k+1}:

$$c_k-\frac{1}{2}\gamma_k = c_{k+1}(1+\gamma_k\beta_k+2L_0\epsilon\gamma_k)+(2L^2+L_0^2\epsilon^2)(L\gamma_k^2+2c_{k+1}\gamma_k^2)+\frac{\beta_k}{2}L_0\epsilon\gamma_k$$

For convenience, define $A_k=1+\gamma_k\beta+2L_0\epsilon\,\gamma_k+(4L^2+2L_0^2\epsilon^2)\gamma_k^2$, $B_k=(2L^2+L_0^2\epsilon^2)L\gamma_k^2+\frac{\beta}{2}L_0\epsilon\,\gamma_k$. Then, the recurrence can be rewritten as $c_k=\frac{1}{2}\gamma_k+A_k\,c_{k+1}+B_k$. Next, we use induction to prove that for any $\eta>0$, there exists a sequence of $\{\gamma_k\}_{k=1}^m$ (depending only on L, L_0, ϵ, and β) such that for all indices k we have $c_k\leq\eta L_0\epsilon$.

Since we start from $c_m = 0$, we have $c_m = 0 \leq \eta_0 \, L_0\epsilon$ holding for any positive $\eta_0 < \eta$. Next, assume for some $k+1$, $c_{k+1} \leq \eta_{k+1} \, L_0\epsilon$ where $\eta_0 <= \eta_k < \eta$. Then we can work out c_k as:

$$c_k = \frac{1}{2}\gamma_k + A_k \, c_{k+1} + B_k \leq \frac{1}{2}\gamma_k + A_k\Big(\eta_k \, L_0\epsilon + C \, \gamma_k\Big) + B_k.$$

Since we can adjust γ_k to be small, we can expand $A_k = 1+\gamma_k\beta+2L_0\epsilon\,\gamma_k+\mathcal{O}(\gamma_k^2)$ and $B_k = \frac{\beta}{2}L_0\epsilon\,\gamma_k+\mathcal{O}(\gamma_k^2)$. Thus,

$$\begin{aligned} c_k &\leq \frac{1}{2}\gamma_k + \Big(1 + \gamma_k\beta + 2L_0\epsilon\,\gamma_k\Big)\eta_{k+1} \, L_0\epsilon + \frac{\beta}{2}L_0\epsilon\,\gamma_k + C_1\gamma_k^2 \\ &= \eta_{k+1} \, L_0\epsilon + \left\{\frac{1}{2} + \eta_{k+1} \, L_0\epsilon(\beta + 2L_0\epsilon) + \frac{\beta}{2}L_0\epsilon\right\}\gamma_k + C_1\gamma_k^2 \\ &= \eta_{k+1} \, L_0\epsilon + C_0\gamma_k + C_1\gamma_k^2. \end{aligned}$$

where C_0, C_1 are constants depending on L, L_0, ϵ, β. Fix any sequence $\{\omega_k\}_{k=0}^{m-1}$ such that $\sum_{k=0}^{m-1} \omega_k < \eta - \eta_0$. Then by choosing $\gamma_k = \min\Big\{1, \ \frac{\omega_k}{C_0+C_1}\Big\}$, we obtain $c_k \leq (\eta_{k+1} + \omega_k)L_0\epsilon$. This completes the proof.

□

Next, we prove that with the learning rates in Lemma D.1, then $\Gamma > 0$. We first assume $\gamma_k < \frac{1}{8L+\frac{8}{3}L_0\epsilon}$.

$$\Gamma_k = \gamma_k - \frac{c_{k+1}\gamma_k + \frac{1}{2}L_0\epsilon\gamma_k}{\beta} - 2L\gamma^2 - 4c_{k+1}\gamma_k^2.$$

Using the bound $c_{k+1} \leq \eta L_0\epsilon$ and setting $\beta = 4L + \frac{4}{3}L_0\epsilon$, we obtain

$$\begin{aligned} \Gamma_k &\geq \gamma_k - \frac{(\eta L_0\epsilon + \frac{1}{2}L_0\epsilon)\gamma_k}{4L + \frac{4}{3}L_0\epsilon} - 2L\gamma_k^2 - 4\eta L_0\epsilon\gamma_k^2 \\ &= \gamma_k - \frac{(\eta + \frac{1}{2})L_0\epsilon\gamma_k}{4L + \frac{4}{3}L_0\epsilon} - 2L\gamma_k^2 - 4\eta L_0\epsilon\gamma_k^2. \end{aligned}$$

Then we can bound the quadratic terms as follows:

- $2L\gamma_k^2 \leq 2L\left(\frac{1}{8L+\frac{8}{3}L_0\epsilon}\right)\gamma_k < 2L \cdot \frac{1}{8L}\gamma_k = \frac{\gamma_k}{4}$
- $4\eta L_0\epsilon\gamma_k^2 \leq \frac{2\eta L_0\epsilon\gamma_k}{4L+\frac{4}{3}L_0\epsilon}$ (since $\gamma_k \leq \frac{1}{8L+\frac{8}{3}L_0\epsilon}$).

Thus,

$$\Gamma_k \geq \gamma_k - \frac{(\eta + \frac{1}{2})L_0\epsilon\gamma_k}{4L + \frac{4}{3}L_0\epsilon} - \frac{\gamma_k}{4} - \frac{2\eta L_0\epsilon\gamma_k}{4L + \frac{4}{3}L_0\epsilon}.$$

Combine the two fractions involving $L_0\epsilon$:

$$\Gamma_k \geq \gamma_k\left[1 - \frac{1}{4} - \frac{(3\eta + \frac{1}{2})L_0\epsilon}{4L + \frac{4}{3}L_0\epsilon}\right].$$

To bound $\Gamma_k \geq \frac{\gamma_k}{4}$, we need:

$$1 - \frac{1}{4} - \frac{(3\eta + \frac{1}{2})L_0\epsilon}{4L + \frac{4}{3}L_0\epsilon} \geq \frac{1}{4}.$$

This means if $\eta \leq \frac{4L+\frac{1}{3}L_0\epsilon}{6L_0\epsilon}$, we have $\Gamma \geq \frac{\gamma_k}{4}$. Finally, denote $\frac{4L+\frac{1}{3}L_0\epsilon}{6L_0\epsilon}$ as η_0, then consider $\eta = \eta_0$ in Lemma D.1, we can let $\omega_k = \omega = \frac{\eta_0}{2m}$ to ensure $\Sigma_{i=1}^m \omega_k < \eta_0$. Then we can set $\gamma = \min\{1, \frac{1}{8L+\frac{8}{3}L_0\epsilon}\frac{\omega}{C_0+C_1}\}$ and set $\gamma_k = \gamma$ so that Lemma D.1 also holds. Aggregating all these together, we finally prove that $\Gamma \geq \min \frac{\gamma_k}{4} = \frac{\gamma}{4}$ which is a constant, resulting in the $\mathcal{O}(\frac{1}{T})$ convergence given that γ is a constant.

Finally, consider Δ_1 when $\Gamma \geq \frac{\gamma}{4}$, it is obvious it reduces to $L_1^2\epsilon^2 + L_0^2L_1^2\epsilon^4$ multiplied by some constant. When ϵ is smaller than 1, the error neighborhood is $\mathcal{O}(\epsilon^2 + \epsilon^4)$. □

E Proof of Corollary 5.5

Corollary. Assume we have a finite distribution setting where the population consists of n samples in total and $\alpha \in (0, 1)$. Then when we have fixed parameters at each round, i.e., $\gamma_k = \gamma = \mathcal{O}(\frac{1}{n^\alpha}), \beta_k = \beta = \mathcal{O}(n^{\frac{\alpha}{2}}), m = \mathcal{O}(n^{\frac{\alpha}{2}})$, then we have $\Gamma = \mathcal{O}(\frac{1}{n^\alpha})$, and the IFO complexity is $\mathcal{O}(\frac{(n^\alpha+n^{1+\frac{\alpha}{2}})\Delta_0}{\delta})$ to achieve $\mathcal{O}(\delta)$-SPS in addition to the error neighborhood.

Proof. From the Lemma D.1 and the proof in App. D, we already know that when $m = \mathcal{O}(n^{\frac{\alpha}{2}})$, any $\gamma_k \leq \gamma = \mathcal{O}(\frac{\eta_0}{2m(C_0+C_1)})$ will make $\Gamma > \frac{\gamma_k}{2}$. Since C_0, C_1 are all $\mathcal{O}(\beta) = \mathcal{O}(n^{\frac{\alpha}{2}})$, then we know $\gamma = \mathcal{O}(\frac{1}{n^\alpha})$ is a plausible choice to satisfy Lemma D.1 to let $\Gamma > \frac{\gamma}{2}$. Moreover, $\Gamma > \frac{\gamma}{2} = \mathcal{O}(\frac{1}{n^\alpha})$ With this fact, we let $\frac{\Delta_0}{T\Gamma} \leq \delta$ to get T should be $\mathcal{O}(\frac{n^\alpha\Delta_0}{\delta})$. Since for each m rounds (1 epoch), we would have $2m + n$ evaluations of gradients. So the average sample complexity at each round is $2 + \frac{n}{m}$. Then, the final IFO complexity is:

$$T(2 + \frac{n}{m}) = \mathcal{O}\left(\frac{\Delta_0 n^\alpha(2 + n^{1-\frac{\alpha}{2}})}{\delta}\right) = \mathcal{O}(\frac{(n^\alpha + n^{1+\frac{\alpha}{2}})\Delta_0}{\delta})$$

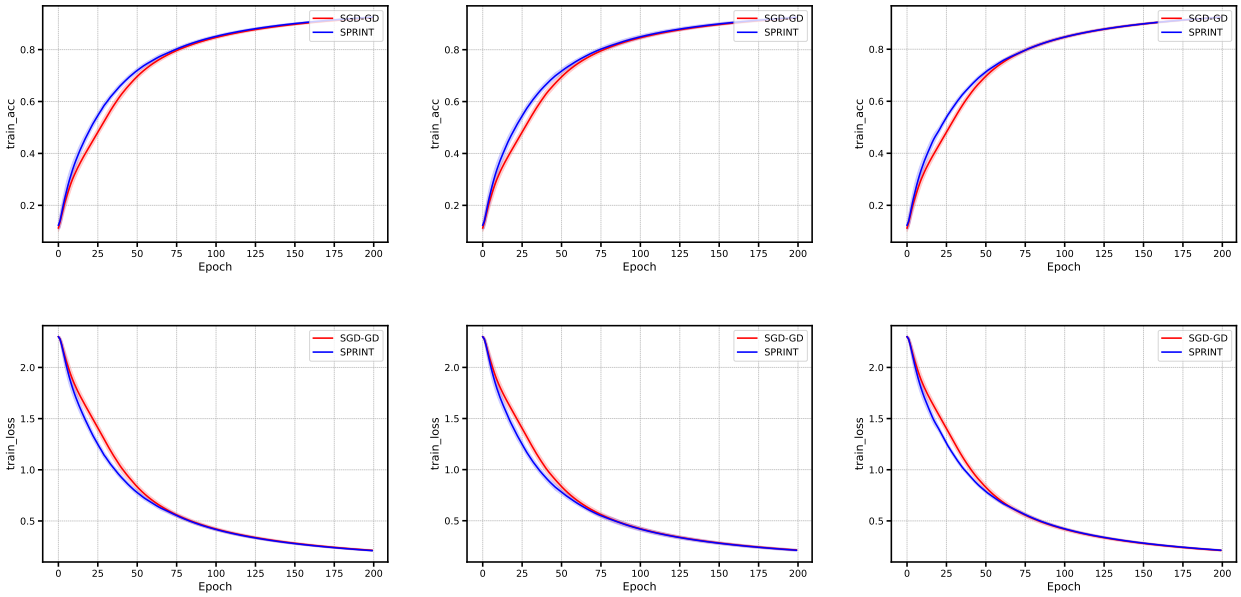
□

F Additional experimental results

Additional experimental settings. We conduct all the experiments on a server which has two Intel Xeon 6326 CPUs and a Nvidia A6000 GPU. We implement our code using the pytorch of version 2.4.1 and seed 2024. In the PP settings, each training requires extensive steps to model distributional shifts, which significantly increases the computational cost, especially on large-scale datasets like CIFAR-10. We are only able to run one experiment on CIFAR-10, but we include experiments using 3 random seeds for the MNIST dataset as follows.

Additional experiments on CIFAR-10 and Credit dataset. We additionally report error bars for Credit and CIFAR-10 (with 10% of the data randomly selected) by rerunning each experiment with three different random seeds. The results are shown in Fig. 5 and Fig. 6. In Fig. 6, the error bars are relatively large compared to the gap between SPRINT and SGD-GD, which is expected since the Credit dataset is relatively small and SGD-GD can already converge fairly quickly. Nevertheless, SPRINT consistently maintains an advantage over SGD-GD across seeds.

Experiments on MNIST dataset. We use the MNIST dataset (Deng, 2012) and train a two-layer MLP to predict the 10 possible digits. We randomly sample approximately 12000 images and train the model using a learning rate set at 0.003. Similar to retention dynamics (Hashimoto et al., 2018; Jin et al., 2024; Zhang et al., 2019), we assume the class distribution will change based on the performance of the current model. The fraction in class c at iteration k of $p_c^{k+1} = \frac{e^{-\alpha\ell_c^k}}{\sum_{c'} e^{-\alpha\ell_{c'}^k}}$, where ℓ_c^k is the loss expectation of class c in iteration k. This means the image class with larger loss now will be less likely to appear at the next



(a) $\alpha = 20$ (b) $\alpha = 50$ (c) $\alpha = 80$

Figure 5: **CIFAR-10** dataset: training accuracy (first row) and training loss (second row) of SGD-GD and SPRINT under different α. Larger α means more intense retention dynamics and the shadow demonstrates standard errors.

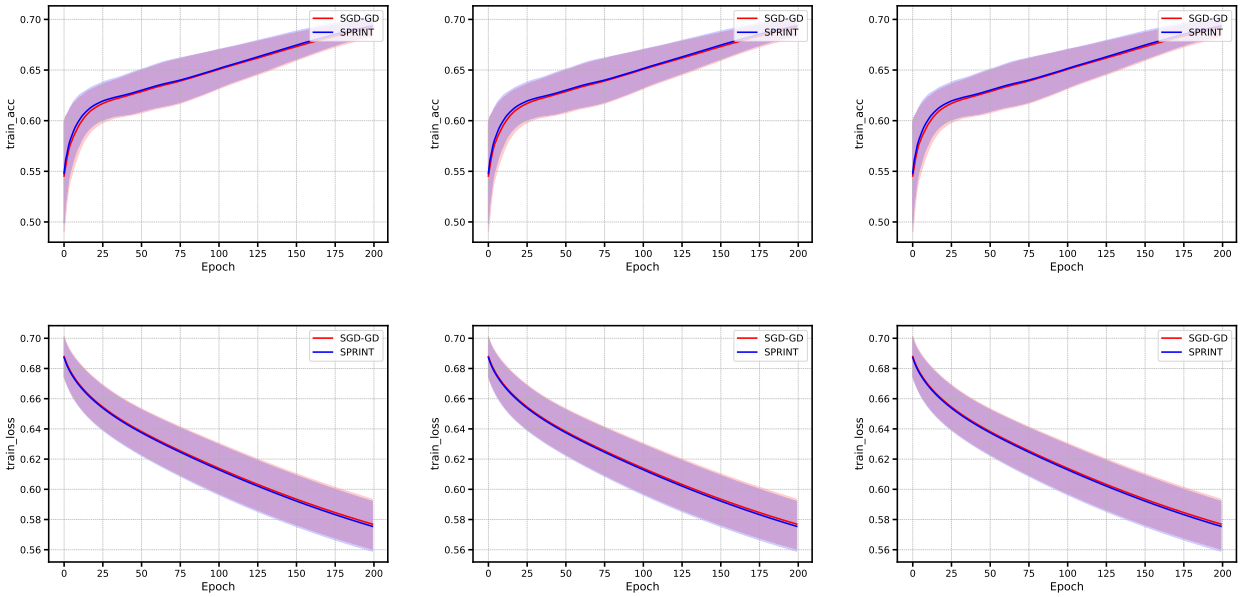
iteration (Zhang et al., 2019). We train the model for 80 epochs with cross entropy loss and visualize the accuracy and training loss curve in Fig. 7. We use three random seeds, which are 2024, 2025, 2026, to run the experiments. The α is set at $20, 50, 80$ to simulate different magnitudes of performative effects. In Fig. 7, the training loss of SPRINT consistently converges faster, which is also reflected in the training accuracy.

Moreover, we also iterate through different learning rates and verify that SPRINT consistently outperforms SGD-GD. In Fig. 9, we plot the dynamics of training loss and accuracy when learning rate in $\{0.001, 0.003, 0.1\}$ and the results demonstrate the competence of SPRINT .

The average of squared gradient norm. An SPS solution is characterized by a small gradient norm $\nabla\|\mathcal{J}(\tilde{\theta}^s, \tilde{\theta}^s)\|^2 \approx 0$. To empirically verify the SPRINT converges to an SPS solution, we track this quantity directly. We visualize the cumulative average of the squared gradient norm at the end of each epoch $\|\nabla\mathcal{J}(\widetilde{\boldsymbol{\theta}}^s; \widetilde{\boldsymbol{\theta}}^s)\|^2$ of SGD-GD and SPRINT in all 3 settings as specified in Sec. 6 shown in Fig. 8. Notably, in MNIST setting (Fig. 8d to Fig. 8f), SPRINT constantly has much smaller squared gradient norm, demonstrating the effectiveness of our algorithm. In Credit setting (Fig. 8a to Fig. 8c), SPRINT constantly has a smaller squared gradient norm when $\alpha = 0.01$ and 0.4. When $\alpha = 0.2$, SPRINT has a much worse initial point than SGD-GD, resulting in the large squared gradient norm at first but similar norm at the end. Similarly, in CIFAR-10 setting (Fig. 8g to Fig. 8i), SPRINT has a smaller squared gradient norm when $\alpha = 50$ and 80. When $\alpha = 20$, SPRINT has a much worse initial point than SGD-GD, resulting in the large squared gradient norm at first but similar norm at the end.

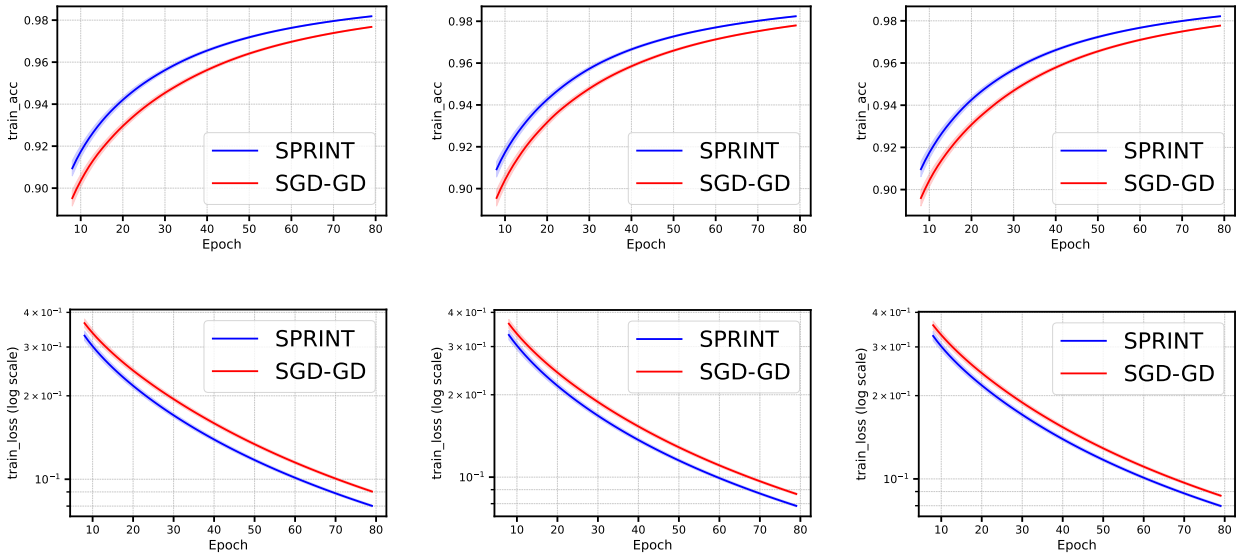
Note that Thm. 5.4 is on the expectation of the squared gradient norm, but we are only able to conduct one experiment for each setting due to the limitation of computational resource. Thus, we can have some bad initialization point of SPRINT to influence the results.

Out-of-Sample Evaluation. To assess whether SPRINT converges to a generalizable solution, we evaluate the out-of-sample evaluation accuracy on a held-out test set for the Credit Dataset and MNIST Dataset under different performativity strengths α. The results are shown in Figure 10. The results demonstrate that SPRINT achieves competitive or superior evaluation accuracy compared to SGD-GD across all settings, indicating that its faster convergence to an SPS solution does not come at the cost of generalization. We note



(a) $\alpha = 0.01$ (b) $\alpha = 0.2$ (c) $\alpha = 0.4$

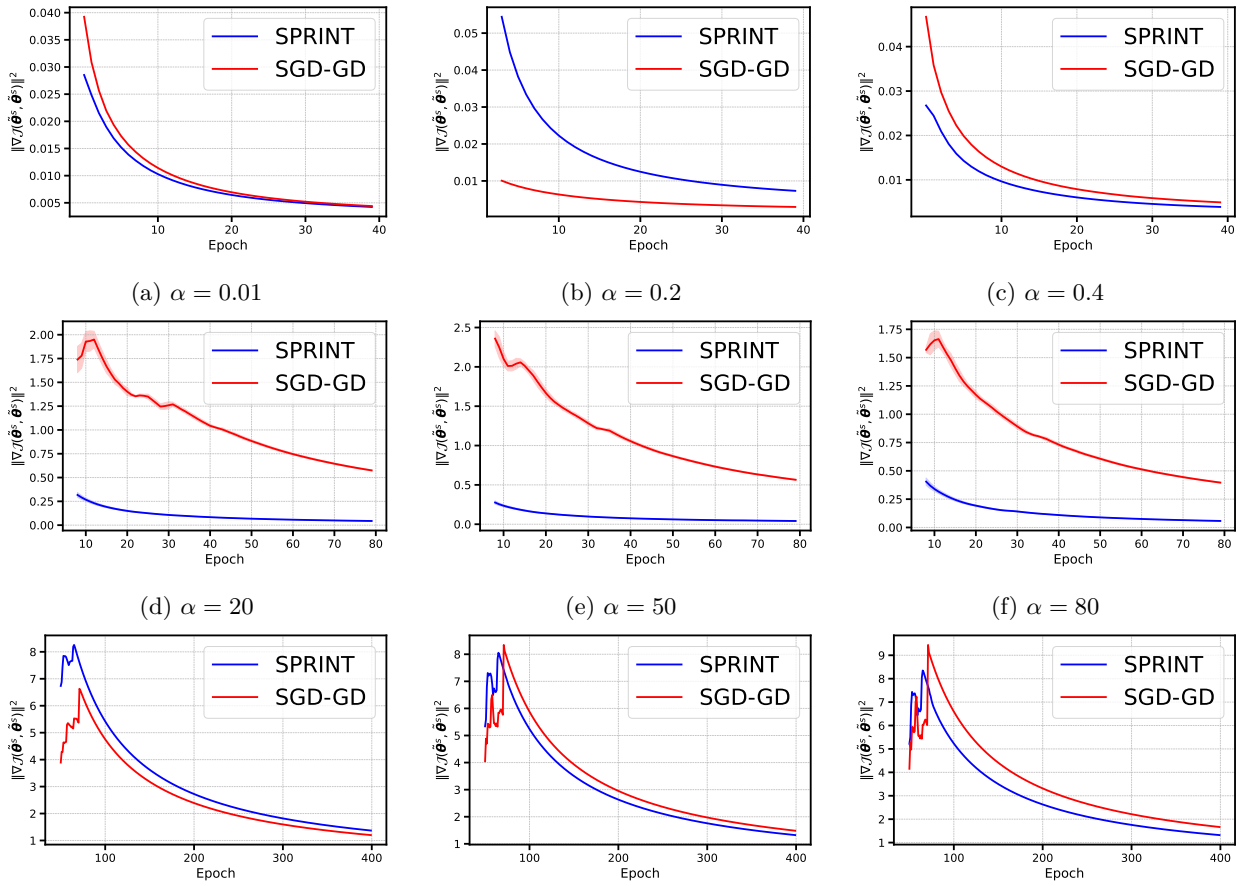
Figure 6: **Credit** dataset: training accuracy (first row) and training loss (second row) of SGD-GD and SPRINT under different α. Larger α means more intense strategic behaviors and the shadow demonstrates standard errors.



(a) $\alpha = 20$ (b) $\alpha = 50$ (c) $\alpha = 80$

Figure 7: **MNIST** dataset: training accuracy (first row) and training loss (second row) of SGD-GD and SPRINT under different α. Larger α means more intense retention dynamics and the shadow demonstrates standard errors.

that evaluating out-of-sample performance in the PP setting is non-trivial: since the data distribution $D(\theta)$ is itself model-dependent, the test distribution shifts as the model evolves. We evaluate on a held-out subset



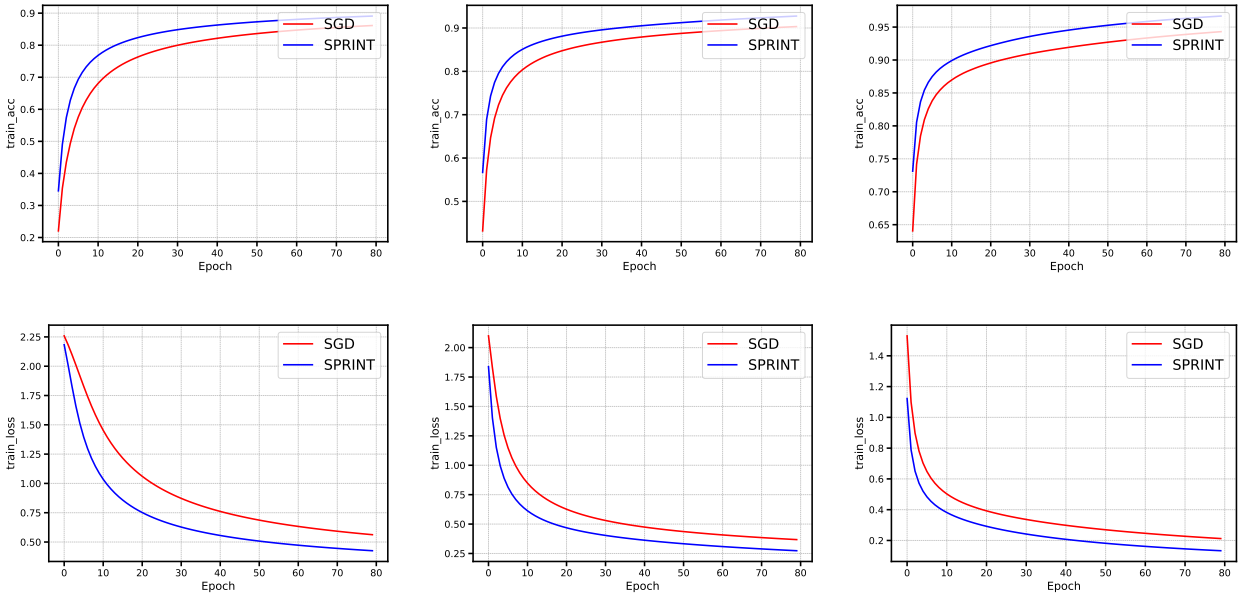
(a) $\alpha = 0.01$ (b) $\alpha = 0.2$ (c) $\alpha = 0.4$

(d) $\alpha = 20$ (e) $\alpha = 50$ (f) $\alpha = 80$

(g) $\alpha = 20$ (h) $\alpha = 50$ (i) $\alpha = 80$

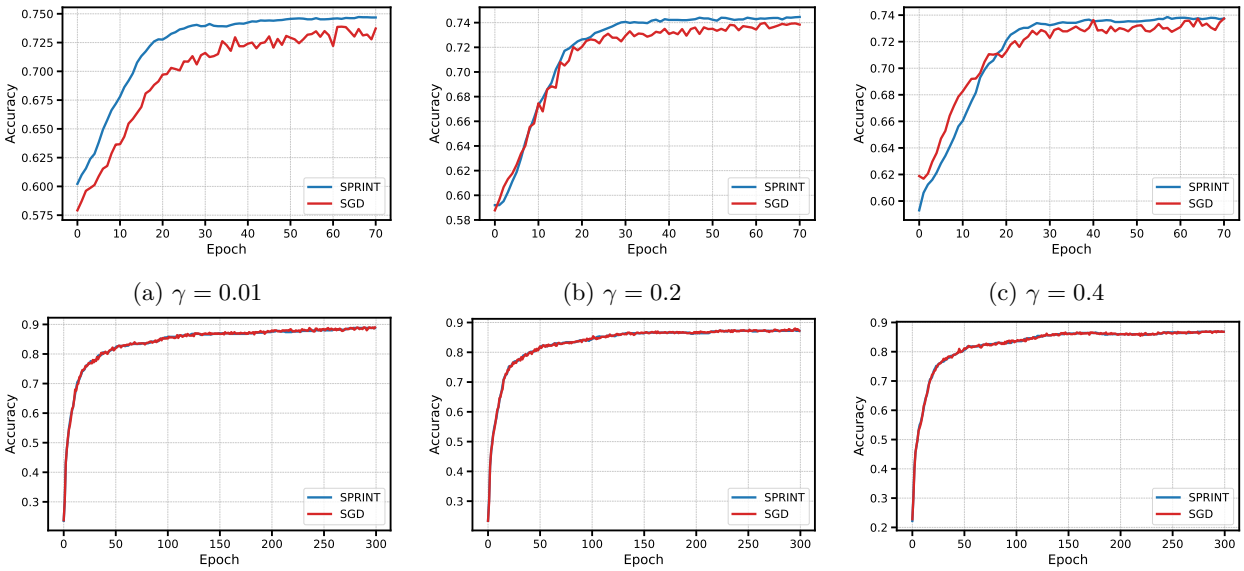
Figure 8: Squared gradient norm $\|\nabla \mathcal{J}(\widetilde{\boldsymbol{\theta}}^s; \widetilde{\boldsymbol{\theta}}^s)\|^2$. From Up to down are **Credit** dataset, **MNIST** dataset, and **CIFAR-10** dataset.

drawn from the distribution induced by the final deployed model, which reflects the practical deployment scenario where the model has reached a stable point.



(a) $\gamma = 0.001$ (b) $\gamma = 0.003$ (c) $\gamma = 0.01$

Figure 9: **MNIST** dataset: training accuracy (first row) and training loss (second row) of SGD-GD and SPRINT under different learning rate γ.



(a) $\gamma = 0.01$ (b) $\gamma = 0.2$ (c) $\gamma = 0.4$

(d) $\gamma = 20$ (e) $\gamma = 50$ (f) $\gamma = 80$

Figure 10: Out-of-sample evaluation accuracy of SGD-GD and SPRINT on the **Credit** and **MNIST** datasets under different performativity strengths α. From Up to down are **Credit** dataset and **MNIST** dataset. SPRINT consistently achieves competitive or superior evaluation accuracy compared to SGD-GD, demonstrating that faster convergence to an SPS solution does not come at the cost of generalization.