Second-Order Convergence in Private Stochastic Non-Convex Optimization

Youming Tao
TU Berlin
tao@ccs-labs.org

Zuyuan Zhang
George Washington University
zuyuan.zhang@gwu.edu

Dongxiao Yu
Shandong University
dxyu@sdu.edu.cn

Xiuzhen Cheng
Shandong University
xzcheng@sdu.edu.cn

Falko Dressler
TU Berlin
dressler@ccs-labs.org

Di Wang
KAUST
di.wang@kaust.edu.sa

Abstract

We investigate the problem of finding second-order stationary points (SOSP) in differentially private (DP) stochastic non-convex optimization. Existing methods suffer from two key limitations: **(i)** inaccurate convergence error rate due to overlooking gradient variance in the saddle point escape analysis, and **(ii)** dependence on auxiliary private model selection procedures for identifying DP-SOSP, which can significantly impair utility, particularly in distributed settings. To address these issues, we propose a generic perturbed stochastic gradient descent (PSGD) framework built upon Gaussian noise injection and general gradient oracles. A core innovation of our framework is using model drift distance to determine whether PSGD escapes saddle points, ensuring convergence to approximate local minima without relying on second-order information or additional DP-SOSP identification. By leveraging the adaptive DP-SPIDER estimator as a specific gradient oracle, we develop a new DP algorithm that rectifies the convergence error rates reported in prior work. We further extend this algorithm to distributed learning with arbitrarily heterogeneous data, providing the first formal guarantees for finding DP-SOSP in such settings. Our analysis also highlights the detrimental impacts of private selection procedures in distributed learning under high-dimensional models, underscoring the practical benefits of our design. Numerical experiments on real-world datasets validate the efficacy of our approach.

1 Introduction

Stochastic optimization is a fundamental problem in machine learning and statistics, aimed at training models that generalize well to unseen data using a finite sample drawn from an unknown distribution. As the volume of sensitive data continues to grow, privacy has become a pressing concern. This has led to the widespread adoption of differential privacy (DP) [10], which provides rigorous privacy guarantees while preserving model utility in learning tasks.

In the past decade, significant progress has been made in DP stochastic optimization, particularly for convex objectives [7, 28, 40, 38, 43, 42]. While convex problems are relatively well understood, non-convex optimization introduces unique challenges, primarily due to the presence of saddle points. Most existing DP algorithms for non-convex problems focus on finding first-order stationary points (FOSP), characterized by small gradient norms [1, 4, 56, 55]. However, FOSP include not only local minima but also saddle points and local maxima, often leading to suboptimal solutions [20, 41]. Consequently, second-order stationary points (SOSP), where the gradient is small and the Hessian is positive semi-definite, are more desirable as they guarantee convergence to local minima.

Motivated by this, substantial research has been devoted to finding SOSP in non-convex optimization [13, 23, 9, 21, 16]. However, the study of SOSP under differential privacy constraints (DP-SOSP) remains limited. At the same time, distributed learning has become increasingly important for training large-scale models across decentralized edge devices. Yet, no existing work has addressed DP-SOSP in non-convex stochastic optimization under distributed settings. Compared to single-machine setups, distributed learning introduces additional challenges, including data heterogeneity, cross-participant privacy, and communication efficiency.

Limitations in the State-of-the-Art. A notable exception in the study of DP-SOSP for stochastic optimization is the recent work by [29], which injects additional Gaussian noise into the DP gradient estimator near saddle points to facilitate escape. Despite its contributions, this method suffers from two key limitations. **(i)** Its saddle point escape analysis overlooks the variance of gradients, leading to incorrect error bounds. A direct correction of the analysis would unfortunately yield a weaker type of SOSP guarantee than originally targeted. This is because their design relies on additional injected noise beyond the inherent DP noise for escape, highlighting the need for an effective way of exploiting the DP noise already present. **(ii)** Their learning algorithm outputs all model iterates and guarantees only the *existence* of a DP-SOSP, requiring an auxiliary private model selection procedure to identify one. While effective in single-machine settings, it faces critical issues in distributed environments due to decentralized data access. In particular, auxiliary private selection introduces non-negligible error and communication overhead, especially when sharing high-dimensional second-order information. These drawbacks also underscore the necessity of a new learning algorithm that inherently outputs a DP-SOSP without dependence on any additional private selection procedure.

Our Contributions. We refer the reader to Appendix 3 for more detailed discussions of the limitations outlined above. To address the challenges identified above, we propose a generic algorithmic and analytical framework for finding DP-SOSP in stochastic non-convex optimization. Our approach not only corrects existing error rates but also extends naturally to distributed learning. The main contributions are summarized as follows:

1. A generic non-convex stochastic optimization framework: We introduce a perturbed stochastic gradient descent (PSGD) framework that employs Gaussian noise and general stochastic gradient oracles. This framework serves as a versatile optimization tool for non-convex stochastic problems beyond the DP setting. A key innovation is a novel criterion based on model drift distance, which enables provable saddle point escape and guarantees convergence to approximate local minima with low iteration complexity and high probability.

2. Corrected error rates for DP non-convex optimization: By incorporating the adaptive DP-SPIDER estimator as the gradient oracle, we develop a differentially private algorithm that achieves a corrected error rate bound of $\tilde{O}\big(\frac{1}{n^{1/3}} + \big(\frac{\sqrt{d}}{\epsilon n}\big)^{2/5}\big)$, where n is the number of samples. This corrects the suboptimal bound of $\tilde{O}\big(\frac{1}{n^{1/3}} + \big(\frac{\sqrt{d}}{\epsilon n}\big)^{3/7}\big)$ reported in [29].

3. Application to distributed learning: We extend the adaptive DP-SPIDER estimator to distributed learning. Via adaptivity, our learning algorithm improves upon the DIFF2 [36], which only guarantees convergence to DP-FOSP under *homogeneous* data. In contrast, our method provides the first error bound for converging to DP-SOSP under arbitrarily *heterogeneous* data: $\tilde{O}\big(\frac{1}{(mn)^{1/3}} + \big(\frac{\sqrt{d}}{\epsilon mn}\big)^{2/5}\big)$, where m is the number of participants and n is the number of samples per participant. Furthermore, we analyze the adverse effects of private model selection, showing that it deteriorates utility guarantees in high-dimensional regimes, thereby highlighting the necessity of our framework.

Due to the space limit, **literature review**, **technical lemmata**, **further discussions**, **omitted proofs**, **experimental results**, **broader impacts** and **conclusions** are all included in the Appendix.

2 Related Work

Private Stochastic Optimization Differential privacy (DP) has become a crucial consideration in stochastic optimization due to increasing concerns about data privacy. The pioneering work by [10] established the foundational principles of DP, and its application in stochastic optimization has since seen significant progress. Early efforts primarily focused on convex optimization, achieving strong privacy guarantees while ensuring efficient learning, with a long list of representative works e.g., [5, 51, 48, 3, 47, 49, 14, 4, 19, 43, 40, 7, 39]. Recent advances have extended DP to non-convex settings, mainly focusing on first-order stationary points (FOSP). Notable works in this area include [46, 56, 4, 53, 1], which improved error rates in non-convex optimization with balanced privacy and utility in stochastic gradient methods. However, these works generally fail to address the more stringent criterion of second-order stationary points (SOSP). The very recent work [29] tired to narrow this gap, but unfortunately has some issues in their results as we discussed before. Our work builds on this foundation by correcting error rates and proposing a framework that ensures convergence to SOSP while maintaining DP.

Finding Second-Order Stationary Points (SOSP) In non-convex optimization, convergence to FOSP is often insufficient, as saddle points can lead to sub-optimal solutions [20, 41]. Achieving SOSP, where the gradient is small and the Hessian is positive semi-definite, ensures that the optimization converges to a local minimum rather than a saddle point. Techniques for escaping saddle points, such as perturbed SGD with Gaussian noise, have been explored in works like [16] and [23]. [16] first showed that SGD with a simple parameter perturbation can escape saddle points efficiently. Later, the analysis was refined by [21, 23]. Recently, variance reduction techniques have been applied to second-order guaranteed methods [17, 27].These methods ensure escape from saddle points by introducing noise to the gradient descent process. In contrast, the studies of SOSP under DP are quite limited, and most of them only consider the empirical risk minimization objective, such as [46, 50, 2]. Very recently, [29] addressed the population risk minimization objective, but with notable gaps in their error analysis, particularly in the treatment of gradient variance. Moreover, all of these works are limited to the single-machine setting and cannot be directly extended to the more general distributed learning setting.

Distributed Learning With the rise of large-scale models and decentralized data, distributed learning has gained significant attention. Methods like federated learning [33] have enabled multiple clients to collaboratively train models without sharing their local data. Recent studies, such as [15, 52, 31, 32] have investigated DP learning in distributed settings, but these works are limited to first-order optimality. While some studies have investigated SOSP in distributed learning, their focus was primarily on Byzantine-fault tolerance [54], and communication efficiency [35, 6]. No effort, to our knowledge, has been made to ensure DP-SOSP in distributed learning scenarios with heterogeneous data. Our proposed framework fills this gap by introducing the first distributed learning algorithm with DP-SOSP guarantees while effectively handling arbitrary data heterogeneity across clients.

3 Limitations of the State-of-the-Art

3.1 Limitation 1: Flawed Error Rate Analysis

Gradient variance overlooked in saddle point escape. The error rate bound for finding a DP-SOSP in [29] is fundamentally incorrect. Their analysis relies on Lemma 3.4 therein (adapted from [46, Lemma 12]), which claims that adding Gaussian noise at the same scale as the DP gradient estimation error suffices to reduce the function value with high probability, enabling escape from saddle points. This argument critically depends on proving that the region around a saddle point where SGD may get stuck is sufficiently narrow. Under this condition, perturbation along the escape direction ensures that the SGD sequence can escape with high probability.

However, the analysis neglects a key factor, which is the stochastic gradient variance. Their proof implicitly uses exact gradients of the population risk, which are unavailable to the algorithm. This is evidenced by the equation preceding equation (39) in [46]. Another indication of this oversight is their choice of step size $\eta = 1/M$. While valid for gradient descent with exact gradients, prior work [23] has shown that stochastic gradients require a smaller step size. The use of $\eta = 1/M$ in [29] for population risk minimization reflects a failure to account for gradient stochasticity. This leads to an underestimated gradient complexity and an overestimated effective sample size per gradient estimate, which ultimately results in an overly optimistic error rate. A correct analysis must acknowledge that stochastic gradients increase estimation error, implying that the true error rate for finding a DP-SOSP is weaker than the one reported.

Fixing the proof is insufficient, a new algorithm is necessary. Although the analytical error can be identified, correcting the proof alone does not yield a satisfactory result. Any direct correction would only achieve a weaker $(\alpha, \alpha^{2/5})$-SOSP guarantee, rather than the desired α-SOSP. In particular, the second-order accuracy would degrade to $\widetilde{O}(\alpha^{2/5})$ instead of the ideal $\widetilde{O}(\alpha^{1/2})$.

This limitation arises because the algorithm in [29] can be viewed as a special case of perturbed gradient descent with bounded gradient inexactness as developed in [54], where the DP noise contributes to the perturbation. By invoking [54, Theorem 3], one only obtains an error rate bound with respect to a weaker class of SOSP where the second-order accuracy depends on $\widetilde{O}(\alpha^{2/5})$.

The underlying reason is that both [54] and [30] rely on injecting additional noise to facilitate escape from saddle points, without considering the role of inherent DP Gaussian noise in the gradients. The excessive injected noise degrades the SOSP guarantee.

To fully resolve this issue, a new algorithmic design is required. In the setting of [54], where gradient perturbations stem from adversarial attacks, such degradation is unavoidable since the perturbations can hinder rather than assist escape. However, in the DP setting, the Gaussian noise is well-behaved and can naturally aid saddle point escape. By leveraging the inherent DP noise, it becomes possible to avoid the need for additional injected noise and to achieve α-SOSP convergence as desired. Therefore, relying on the algorithmic designs of [54] or [30] is insufficient, and a new algorithm must be developed to achieve the desired guarantees.

3.2 Limitation 2: Challenges of Private SOSP Selection

Inapplicability of AboveThreshold in distributed learning. The algorithm in [29] guarantees only the existence of an α-SOSP among its iterates. To privately identify such a point, it applies the AboveThreshold mechanism to test whether candidate models satisfy the SOSP conditions by privately evaluating gradient norms and Hessian eigenvalues. While this procedure introduces negligible error in single-machine settings, it faces fundamental challenges in distributed learning.

According to [29, Lemma 4.5], for any $x \in \mathbb{R}^d$ and a dataset S of size $O(n)$, with probability at least $1-\omega$, the following holds:

$$\|\nabla F_{\mathcal{D}}(x) - \nabla \hat{f}_S(x)\| \leq O\left(\frac{G\log(d/\omega)}{\sqrt{n}}\right), \quad \|\nabla^2 F_{\mathcal{D}}(x) - \nabla^2 \hat{f}_S(x)\|_{\text{op}} \leq O\left(\frac{M\log(d/\omega)}{\sqrt{n}}\right).$$

This implies:

$$\|\nabla \hat{f}_S(x)\| \leq \|\nabla F_{\mathcal{D}}(x)\| + O\left(\frac{G\log\frac{d}{\omega}}{\sqrt{n}}\right), \lambda_{\min}(\nabla^2 \hat{f}_S(x)) \geq \lambda_{\min}(\nabla^2 F_{\mathcal{D}}(x)) - O\left(\frac{M\log\frac{d}{\omega}}{\sqrt{n}}\right).$$

With these bounds, AboveThreshold can identify a DP-SOSP by setting appropriate thresholds. However, this procedure relies on centralized access to the dataset S.

In distributed learning, each client holds a local dataset S_i. To estimate global quantities, aggregation is required:

$$\|\nabla \hat{f}_S(x)\| \leq \frac{1}{m}\sum_{i=1}^{m} \|\nabla \hat{f}_{S_i}(x)\|, \quad \lambda_{\min}(\nabla^2 \hat{f}_S(x)) \geq \frac{1}{m}\sum_{i=1}^{m} \lambda_{\min}(\nabla^2 \hat{f}_{S_i}(x)).$$

Yet the learning algorithm guarantees only:

$$\|\nabla F_{\mathcal{D}}(x)\| \leq \frac{1}{m}\sum_{i=1}^{m} \|\nabla F_{\mathcal{D}_i}(x)\|, \quad \lambda_{\min}(\nabla^2 F_{\mathcal{D}}(x)) \geq \frac{1}{m}\sum_{i=1}^{m} \lambda_{\min}(\nabla^2 F_{\mathcal{D}_i}(x)),$$

This relationship does not provide an upper bound on $\|\nabla \hat{f}_S(x)\|$ or a lower bound on $\lambda_{\min}(\nabla^2 \hat{f}_S(x))$ solely from local empirical estimates. Therefore, it is infeasible to determine valid thresholds for AboveThreshold based only on local information. Any attempt to perform this selection would require clients to share their (noisy) gradients and Hessians with the server, which introduces substantial privacy, communication, and computation costs.

Eliminating private model selection is essential in distributed learning. A feasible method for private model selection in distributed learning would extend the centralized algortihm of [46, Algorithm 5]. Specifically, each client privately computes gradients and Hessians on additional local data beyond the training set, and the server aggregates these to estimate global quantities. However, this strategy has several drawbacks. It requires extra data outside the training process, increases communication overhead by transmitting high-dimensional gradients and Hessians, and incurs high computational costs. It also shifts the method from a first-order to a second-order algorithm.

Moreover, as shown in Section 7, sharing perturbed high-dimensional gradients and Hessians, rather than one-dimensional scalar queries as in AboveThreshold, introduces non-negligible additional error. This error accumulation degrades the accuracy guarantees provided by the learning algorithm. Unlike the single-machine case, private model selection in distributed learning incurs significant costs in accuracy, privacy, computation, and communication.

These challenges demonstrate the necessity of designing an algorithm that inherently outputs a DP-SOSP without relying on a private model selection procedure. Such a design avoids additional data consumption, computational burden, communication overhead, and deterioration of error guarantees.

4 Preliminaries

Notations. We denote by $\|\cdot\|$ the ℓ_2 norm and by $\lambda_{\min}(\cdot)$ the smallest eigenvalue of a matrix. The symbol $\mathbf{I}_d$ represents the d-dimensional identity matrix. We use $O(\cdot)$ and $\Omega(\cdot)$ to hide constants independent of problem parameters, while $\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ additionally hide polylogarithmic factors.

Stochastic Optimization. Let $f : \mathbb{R}^d \times \mathcal{Z} \to \mathbb{R}$ be a (potentially non-convex) loss function, where $x \in \mathbb{R}^d$ denotes the d-dimensional model parameter and $z \in \mathcal{Z}$ is a data point.

Assumption 1. The loss function $f(\cdot; z)$ is G-Lipschitz, M-smooth, and ρ-Hessian Lipschitz. Specifically, for any $z \in \mathcal{Z}$ and any $x_1, x_2 \in \mathbb{R}^d$, we have: (i) $|f(x_1; z) - f(x_2; z)| \leq G\|x_1 - x_2\|$; (ii) $\|\nabla f(x_1; z) - \nabla f(x_2; z)\| \leq M\|x_1 - x_2\|$; (iii) $\|\nabla^2 f(x_1; z) - \nabla^2 f(x_2; z)\| \leq \rho\|x_1 - x_2\|$.

Let $\mathcal{D}$ denote the unknown data distribution. The population risk is defined as the *expected* loss: $F_{\mathcal{D}}(x) := \mathbb{E}_{z\sim\mathcal{D}}[f(x; z)]$ for $\forall x \in \mathbb{R}^d$. When clear from context, we omit $\mathcal{D}$ and simply write $F(x)$.

Assumption 2. Let x^* denote a minimizer of the population risk and $F^* = F(x^*)$ its minimum value. There exists $U \in \mathbb{R}$ such that $\max_x F(x) - F^* \leq U$.

Let D denote a dataset of n i.i.d. samples from $\mathcal{D}$. The empirical risk is defined as $\hat{f}_D(x) := \frac{1}{|D|}\sum_{z\in D} f(x; z)$. Given access to D, the goal is to find an approximate second-order stationary point (SOSP) of the unknown population risk $F(\cdot)$. In general, we have the notion of (α_g, α_H)-SOSP:

Definition 1 ((α_g, α_H)-SOSP)**.** A point x is an (α_g, α_H)-SOSP of a twice differentiable function $F(\cdot)$ if x satisfies $\|\nabla F(x)\| \leq \alpha_g$ and $\nabla^2 F(x) \succeq -\alpha_H \cdot \mathbf{I}_d$.

As shown in [54, Proposition 1], there exists a lower bound of $\tilde{O}(\alpha_g^{1/2})$ for α_H given α_g, implying that an $(\alpha, \tilde{O}(\sqrt{\alpha}))$-SOSP is the best second-order guarantee achievable. Accordingly, we target the notion of α-SOSP in this work, following [29].

Definition 2 (α-SOSP)**.** A point x is an α-SOSP of a twice differentiable function $F(\cdot)$ if x satisfies $\|\nabla F(x)\| \leq \alpha$ and $\nabla^2 F(x) \succeq -\sqrt{\rho\alpha} \cdot \mathbf{I}_d$.

An α-SOSP excludes α-strict saddle points where $\nabla^2 F(x) \preceq -\sqrt{\rho\alpha}\mathbf{I}_d$, thereby ensuring convergence to an approximate local minimum. Following prior work [29, 23], we assume $M \geq \sqrt{\rho\alpha}$ so that finding an SOSP is strictly more challenging than finding an FOSP.

Distributed Learning. In the distributed (federated) learning setting, m clients collaboratively learn under the coordination of a central server. Each client $j \in [m]$ has a local dataset D_j of size n, sampled from an unknown local distribution $\mathcal{D}_j$. The population risk for client j is defined as $F_{\mathcal{D}_j}(x) \coloneqq \mathbb{E}_{z \sim \mathcal{D}j}[f(x; z)]$ or simply $F_j(x)$. The global population risk is defined as the average of the local population risks: $F_{\mathcal{D}}(x) \coloneqq \frac{1}{m}\sum_{j\in[m]} F_j(x)$, or simply $F(x)$. We allow for heterogeneous local datasets, meaning that the local distributions $\{\mathcal{D}_j\}_{j\in[m]}$ may differ arbitrarily.

Differential Privacy. We aim to find an α-SOSP under the requirment of Differential Privacy (DP), which is referred to as an α-DP-SOSP. We say two datasets D and D' are *adjacent* if they differ by at most one record. DP ensures that the output of the stochastic optimization algorithm on any pair of adjacent datasets is statistically indistinguishable.

Definition 3 (Differential Privacy (DP) [10])**.** Given $\epsilon, \delta > 0$, a randomized algorithm $\mathcal{A} : \mathcal{Z} \to \mathcal{X}$ is (ϵ, δ)-DP if for any pair of adjacent datasets $D, D' \subseteq \mathcal{Z}$, and any measurable subset $S \subseteq \mathcal{X}$,

$$\mathbb{P}[\mathcal{A}(D) \in S] \leq \exp(\epsilon) \cdot \mathbb{P}[\mathcal{A}(D') \in S] + \delta.$$

In distributed learning, we focus on *inter-client record-level DP (ICRL-DP)*, which assumes that clients do not trust the server or other clients with their sensitive local data. This notion has been widely adopted in state-of-the-art distributed learning works, such as [15, 31, 32].

Definition 4 (Inter-Client Record-Level DP (ICRL-DP))**.** Given $\epsilon, \delta > 0$, a randomized algorithm $\mathcal{A} : \mathcal{Z}^m \to \mathcal{X}$ satisfies (ϵ, δ)-ICRL-DP if, for any client $j \in [m]$ and any pair of local datasets D_j and D'_j, the full transcript of client j's sent messages during the learning process satisfies (3), assuming fixed local datasets for other clients.

Variance Reduction via SPIDER. Since the population risk $F(\cdot)$ is unknown, standard SGD approximates the true gradient $\nabla F(x_{t-1})$ at iteration t using a stochastic estimate g_t. However, such estimates often exhibit high variance, degrading convergence. The Stochastic Path Integrated Differential Estimator (SPIDER) [12] mitigates this variance using two gradient oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. For a mini-batch $\mathcal{B}_t$ at iteration t, we define

$$\mathcal{O}_1(x_{t-1}, \mathcal{B}_t) \coloneqq \nabla \hat{f}_{\mathcal{B}_t}(x_{t-1}), \quad \mathcal{O}_2(x_{t-1}, x_{t-2}, \mathcal{B}_t) \coloneqq \nabla \hat{f}_{\mathcal{B}_t}(x_{t-1}) - \nabla \hat{f}_{\mathcal{B}_t}(x_{t-2}).$$

SPIDER queries $\mathcal{O}_1$ every p iterations to refresh the gradient estimate. Between these updates, it uses $\mathcal{O}_2$ to incrementally refine the estimate:

$$g_t = \begin{cases} \mathcal{O}_1(x_{t-1}, \mathcal{B}_t), & \text{if } (t-1) \bmod p = 0, \\ g_{t-1} + \mathcal{O}_2(x_{t-1}, x_{t-2}, \mathcal{B}_t), & \text{otherwise.} \end{cases}$$

For smooth functions, the variance of $\mathcal{O}_2(x_{t-1}, x_{t-2}, \mathcal{B}_t)$ scales with $\|x_{t-1} - x_{t-2}\|$, which is typically small when updates are minimal. This allows SPIDER to achieve low-variance gradient estimates while maintaining accuracy.

We choose SPIDER because it achieves state-of-the-art error rates for privately finding first-order stationary points (DP-FOSP) [1]. Our goal is to investigate whether its variance reduction can extend to DP-SOSP. Importantly, the insights in this paper are not specific to SPIDER; they also apply to other variance-reduced methods such as STORM [8] or SARAH [37]. However, since these algorithms are conceptually similar, no significant improvement is expected from substituting them.

Algorithm 1: `Gauss-PSGD`: Gaussian Perturbed Stochastic Gradient Descent

```
Input: Failure probability $\omega$, initial model $x_0$, learning rate $\eta$, # of escape repeats
       $Q$, model deviation threshold $\mathcal{R}$, # of escape steps $\Gamma$
$t \leftarrow 0$;
while true do
    $t \leftarrow t + 1$;
    $\hat{g}_t \leftarrow$ P_Grad_Oracle($*$);
    if $\|\hat{g}_t\| \leq 3\chi$ then
        /* Saddle point escape                                    */
        $\tilde{t} \leftarrow t$, $\tilde{x} \leftarrow x_{t-1}$, esc $\leftarrow$ false;
        for $q \leftarrow 1, \cdots, Q$ do
            $t \leftarrow \tilde{t}$, $x_t \leftarrow \tilde{x}$;
            for $\tau \leftarrow 1, \cdots, \Gamma$ do
                $\hat{g}_t \leftarrow$ P_Grad_Oracle($*$) ;
                $x_t \leftarrow x_{t-1} - \eta \cdot \hat{g}_t$;
                if $\|x_t - \tilde{x}\| \geq \mathcal{R}$ then
                    esc $\leftarrow$ true;
                    break;
                else
                    $t \leftarrow t + 1$;
            if esc $=$ true then
                break;
        if esc $=$ false then
            return $x_{t-1}$
    else
        /* Normal descent step                                    */
        $x_t \leftarrow x_{t-1} - \eta \cdot \hat{g}_t$;
```

5 Our Generic Perturbed SGD Framework

In this section, we introduce a generic framework for finding an α-SOSP of the population risk $F_{\mathcal{D}}(\cdot)$ by escaping saddle points. Our framework is a Gaussian perturbed stochastic gradient descent method, denoted as `Gauss-PSGD`.

5.1 Gradient Oracle Setup

Since $\nabla F_{\mathcal{D}}(\cdot)$ is unknown, direct gradient descent is infeasible. As in standard stochastic optimization, we assume access to a stochastic gradient oracle g_t that approximates $\nabla F_{\mathcal{D}}(x_{t-1})$ at iteration t. For example, g_t can be computed as an empirical gradient over a mini-batch $\mathcal{B}_t$ sampled from $\mathcal{D}$. We model the oracle as

$$g_t = \nabla F(x_{t-1}) + \zeta_t, \tag{1}$$

where ζ_t represents inherent gradient noise. Following [23, 29], we assume $\zeta_t \sim \text{nSG}(\sigma)$, where nSG denotes a norm-sub-Gaussian distribution (Definition 7 in Appendix A).

To enable saddle point escape, we introduce an additional Gaussian perturbation to form a perturbed gradient oracle $\hat{g}_t$:

$$\hat{g}_t = g_t + \xi_t = \nabla F(x_{t-1}) + \zeta_t + \xi_t, \tag{2}$$

where $\xi_t \sim \mathcal{N}(0, r^2 \mathbf{I}_d)$. We define the effective noise magnitude in $\hat{g}_t$ as

$$\psi := \sqrt{\sigma^2 + r^2 d}. \tag{3}$$

The model update is then performed by

$$x_t \leftarrow x_{t-1} - \eta \hat{g}_t. \tag{4}$$

Our problem setting fundamentally differs from that in [23]. In their setting, the target error α is given, and the perturbation magnitude r is determined accordingly. In contrast, in our privacy-constrained setting, r is dictated by the privacy parameters (ϵ, δ), and the goal is to achieve the smallest possible α under this constraint. Crucially, their parameterization $r = O(\sqrt{(\sigma^2 + \alpha^{3/2})/d})$ implies that r depends on both σ and α, determined by $\max\{\sigma/\sqrt{d}, \alpha^{3/4}/\sqrt{d}\}$. This non-invertible relationship between r and α makes their setting incompatible with ours. First, under DP constraints, r is determined by (ϵ, δ) and may be smaller than $\sigma/\sqrt{d}$ in weak privacy regimes, violating the required lower bound. Second, because r and α are not uniquely determined by each other, it is not meaningful to directly translate their error bounds into our setting. Thus, their analysis and results cannot be directly applied to our problem.

5.2 Our Approach: A General Gaussian-Perturbed SGD Framework

We present our `Gauss-PSGD` framework in Algorithm 1. As specified in (2), we employ a general Gaussian-perturbed stochastic gradient oracle, denoted as `P_Grad_Oracle`$(*)$ in steps 4 and 10, where $*$ abstracts the specific arguments required by the oracle implementation. This abstraction allows `Gauss-PSGD` to serve as a flexible optimization framework for non-convex stochastic problems, applicable beyond the differential privacy (DP) setting.

At each iteration, the gradient estimate $\hat{g}_t$ is computed by `P_Grad_Oracle`$(*)$, and the model parameter is updated via the gradient descent step in (4). The algorithm proceeds until it encounters a point $\tilde{x}$ satisfying $\|\hat{g}_t\| \leq 3\chi$, where χ is specified in (5). This point $\tilde{x}$ may lie near a saddle point with a large negative eigenvalue of the Hessian. To escape such a saddle point, the framework enters an escape procedure (steps 6–20), which performs Q rounds of Γ-`descent` (steps 9–16).

In each round, the algorithm executes at most Γ perturbed SGD iterations starting from $\tilde{x}$. If at any iteration we observe $\|x_t - \tilde{x}\| \geq \mathcal{R}$ for a threshold $\mathcal{R}$ (specified in (5)), indicating that the iterate has moved sufficiently far from $\tilde{x}$, we declare that the algorithm has successfully escaped the saddle point and resume normal PSGD from x_t. If no such movement is observed after Q rounds, we declare $\tilde{x}$ an α-SOSP of the population risk $F_{\mathcal{D}}(\cdot)$ and output $\tilde{x}$. The repetition over Q rounds ensures a high probability of escape: as we will prove later, each Γ-`descent` succeeds in escaping a saddle point with constant probability, and multiple repetitions reduce the failure probability to any desired level.

A central innovation of our framework is using model drift distance as the escape criterion (step 12), replacing the function value decrease criterion used in [21, 23]. This design enables the algorithm to identify an SOSP with high probability during the optimization process itself, eliminating the need for an auxiliary private model selection step. Our key insight is as follows: escaping a saddle point not only causes a decrease in the objective function [21, 23] but also induces a substantial displacement of the model parameter beyond a threshold $\mathcal{R}$. Shifting from monitoring function values to tracking parameter movement is critical in population risk settings, where the objective function is unknown and function evaluations are unavailable, unlike in empirical risk minimization [21]. However, the model iterates and their deviations are observable. By leveraging this property, our framework can directly output an SOSP, rather than merely guaranteeing its existence among the iterates.

5.3 Main Results for `Gauss-PSGD` Framework

We begin by introducing the parameter setup and notations used throughout the analysis:

$$\begin{gathered}\iota := s\mu, \quad \chi := 4\sqrt{C}s\mu^2\psi, \quad \alpha := 4\chi, \\ \Gamma := \frac{\iota}{s\eta\sqrt{\rho\alpha}}, \quad \mathcal{R} := \frac{1}{\iota^{1.5}}\sqrt{\frac{\alpha}{\rho}}, \quad \Phi := \frac{s}{8\iota^3}\sqrt{\frac{\alpha^3}{\rho}}, \quad \eta := \frac{\sqrt{\rho\alpha}}{M^2\iota^2}.\end{gathered} \tag{5}$$

where s is a sufficiently large absolute constant to be chosen later, and μ is a logarithmic factor:

$$\mu := \max\left\{\frac{1}{s}\log\left(\frac{9d\log\left(\frac{4C^{1/4}}{s\eta r}\sqrt{\frac{\psi}{\rho}}\right)}{C^{1/4}\eta\sqrt{s\rho\psi}}\right), \log\left(\frac{160\sqrt{2}C^{1/4}}{s\sqrt{\eta r}}\sqrt{\frac{\psi}{\rho}}\right), \frac{\left(C\log\frac{4T}{\omega}\right)^{1/4}}{2^{\frac{3}{4}}\sqrt{s}}, 1\right\}. \tag{6}$$

Here C is an absolute constant that may change across expressions. Let $\tilde{x}$ denote a saddle point of the population risk $F(\cdot)$, and $\mathcal{H} := \nabla^2 F(\tilde{x})$. Let $v_{\min}$ be the eigenvector corresponding to $\lambda_{\min}(\mathcal{H})$, and $\mathcal{P}_{-v_{\min}}$ be the projection onto the orthogonal complement of $v_{\min}$. Set $\gamma := -\lambda_{\min}(\mathcal{H})$.

Definition 5 (Coupling Sequence)**.** Let $\{x_i\}$ and $\{x'_i\}$ be two PSGD sequences initialized at $\tilde{x}$. We say they are *coupled* if they share the same randomness for $\mathcal{P}_{-v_{\min}}\xi_t$ and ζ_t at each iteration t, but use opposite perturbations in the $v_{\min}$ direction: $v_{\min}^\top\xi_t = -v_{\min}^\top\xi'_t$.

The following lemma ensures that under Γ-`descent`, at least one of the coupled sequences escapes the saddle point with constant probability (proof in Appendix B.1).

Lemma 1 (Escaping Saddle Points)**.** Let $\{x_i\}$ and $\{x'_i\}$ be coupled PSGD sequences initialized at $\tilde{x}$ such that $\|\nabla F(\tilde{x})\| \leq \alpha$ and $\lambda_{\min}(\nabla^2 F(\tilde{x})) \leq -\sqrt{\rho\alpha}$. Then, with probability at least 1/4, there exists $\tau \leq \Gamma$ such that $\max\{\|x_\tau - \tilde{x}\|, \|x'_\tau - \tilde{x}\|\} \geq \mathcal{R}$.

From this, we immediately obtain a corollary that applies to any PSGD sequence:

Corollary 1. For any PSGD sequence $\{x_i\}$ starting at $\tilde{x}$ with $\|\nabla F(\tilde{x})\| \leq \alpha$ and $\lambda_{\min}(\nabla^2 F(\tilde{x})) \leq -\sqrt{\rho\alpha}$, with probability at least 1/8, there exists $t \leq \Gamma$ such that $\|x_t - \tilde{x}\| \geq \mathcal{R}$.

To ensure a high-probability escape from a saddle point, we repeat Γ-`descent` for Q rounds:

Lemma 2 (Escape Amplification via Repetition)**.** Given any $\omega_0 \in (0, 1)$, repeating Γ-`descent` independently for $Q = \frac{26}{5}\log(\frac{1}{\omega_0})$ rounds ensures escape with probability at least $1 - \omega_0$.

The proof is deferred to Appendix B.2. We now analyze the total number of PSGD steps needed for convergence. Let $\nu_t := \zeta_t + \xi_t$ denote the combined noise in the gradient estimate.

Lemma 3 (Descent Lemma)**.** For any t_0, the following holds:

$$F(x_{t_0+t}) - F(x_{t_0}) \leq -\frac{\eta}{2}\sum_{i=0}^{t-1}\|\nabla F(x_{t_0+i})\|^2 + \frac{\eta}{2}\sum_{i=1}^{t}\|\nu_{t_0+i}\|^2 \tag{7}$$

Since ν_t can be bounded with high probability, we have:

Corollary 2. For any t_0 and some constant c, with probability at least $1-2e^{-\iota}$,

$$F(x_{t_0+t}) - F(x_{t_0}) \leq -\frac{\eta}{2}\sum_{i=0}^{t-1}\|\nabla F(x_{t_0+i})\|^2 + c\eta\psi^2(t+\iota). \tag{8}$$

Proofs of Lemma 3 and Corollary 2 are in Appendix B.3 and B.4. These imply that large gradients lead to rapid function decrease. We next show in Lemma 4 that a successful saddle point escape via Γ-`descent` leads to a significant decrease in function value, whose proof is in Appendix B.5.

Lemma 4 (Value Decrease per Escape)**.** Let a Γ-`descent` starting from x_{t_0} succeed after $\tau \leq \Gamma$ steps. With probability at least $1-2e^{-\iota}$, $F(x_{t_0+\tau}) - F(x_{t_0}) \leq -\frac{s}{8\iota^3}\sqrt{\frac{\alpha^3}{\rho}} = -\Phi$.

We bound the total number of PSGD steps required for convergence, based on the following estimate:

Lemma 5 (Gradient Estimate Error Bound)**.** With probability at least $1-\omega/2$, for all $t \in [T]$, $\|\nu_t\| \leq C\sqrt{2\log\left(\frac{4T}{\omega}\right)}\psi \leq \chi$.

Lemma 6 (Maximum Number of Descent Steps)**.** Given failure probability ω, set $Q = \frac{26}{5}\log\left(\frac{16\iota^3(F_0-F^*)}{s\omega}\sqrt{\frac{\rho}{\chi^3}}\right)$. `Gauss-PSGD` returns an α-SOSP within at most $\tilde{O}(1/\alpha^{2.5})$ PSGD steps.

Proofs of Lemmas 5 and 6 are in Appendix B.6 and B.7, respectively.

Remark 1 (On Gradient Complexity)**.** While Lemma 6 appears to improve gradient complexity from $O(1/\alpha^4)$ in [23] to $O(1/\alpha^{2.5})$, the two results are not directly comparable. In [23], the error target α is treated as an input and can be arbitrarily small, with gradient variance σ typically treated as a constant. In contrast, in our setting, the perturbation r and variance σ are fixed by privacy constraints, and α emerges as a function of these. Thus, our gradient complexity fundamentally depends on σ and r, though we express it in terms of α for clarity.

Combining all the above, we obtain the final convergence guarantee:

Theorem 1 (Convergence Guarantee of `Gauss-PSGD`)**.** Let Assumptions 1 and 2 hold. For any failure probability $\omega \in (0,1)$, using the parameter settings in (5) and setting $Q = \frac{26}{5}\log\left(\frac{16\iota^3(F_0-F^*)}{s\omega}\sqrt{\frac{\rho}{\chi^3}}\right)$, then with probability at least $1-\omega$, `Gauss-PSGD` (Algorithm 1) returns an α-SOSP of $F(\cdot)$, where $\alpha = 4\chi$, within at most $\tilde{O}(1/\alpha^{2.5})$ PSGD steps.

6 Rectified Error Rate for finding SOSP in DP Stochastic Optimization

6.1 Adaptive Gradient Oracle: Ada-DP-SPIDER

In this section, we derive the upper bound on the error rate for DP stochastic optimization by instantiating the Gauss-PSGD framework with a specific gradient oracle. We adopt an adaptive version of the DP-SPIDER estimator, referred to as Ada-DP-SPIDER, which is presented in Algorithm 2. This adaptive version refines the original SPIDER by dynamically adjusting gradient queries based on model drift. Unlike standard SPIDER, which queries $\mathcal{O}_1$ at fixed intervals and may suffer from growing estimation error over time, Ada-DP-SPIDER tracks the cumulative model drift defined as

$$\mathsf{drift}_t := \sum_{i=\tau(t)}^{t} \|x_i - x_{i-1}\|^2, \tag{9}$$

where $\tau(t)$ is the last iteration at which the full gradient oracle $\mathcal{O}_1$ was queried.

The intuition is that, for smooth functions, the error of $\mathcal{O}_2$, which estimates $\nabla F(x_{t-1}) - \nabla F(x_{t-2})$, is proportional to $\|x_{t-1} - x_{t-2}\|$. When the model drift is small, $\mathcal{O}_2$ remains accurate, allowing for continued use to reduce variance (steps 9-11). However, when the drift becomes large, further use of $\mathcal{O}_2$ can accumulate significant errors. To mitigate this, the algorithm triggers a fresh query to $\mathcal{O}_1$ (steps 4-7). A threshold κ is used in step 3 to determine when the drift is large. This enables adaptive switching between oracles based on the model drift, ensuring the total error remains well controlled.

Our approach differs fundamentally from that of [29]. In their method, in addition to using model drift to trigger $\mathcal{O}_1$, they also invoke $\mathcal{O}_1$ when approaching potential saddle points and inject an additional Gaussian noise on top of the DP gradient estimator to escape. To prevent excessive noise injection, they introduce a Frozen state to restrict how frequently this occurs. In contrast, our method leverages the inherent Gaussian noise from the DP gradient estimator for saddle point escape and uses model drift as the sole trigger for querying $\mathcal{O}_1$. This results in a simpler, more efficient estimator without auxiliary state tracking or redundant noise injection.

6.2 Error Rate Analysis for DP-SOSP with Ada-DP-SPIDER

To minimize the error rate α for DP-SOSP using Ada-DP-SPIDER, we must carefully tune algorithmic parameters, including the mini-batch sizes b_1, b_2, and the drift threshold κ. These parameters directly influence the gradient estimation error, which, according to Theorem 1, dominates the learning error. The following lemma characterizes how these parameters affect the estimation quality:

Lemma 7. Let Assumption 1 hold. For all $t \in [T]$, the gradient estimate $\hat{g}_t$ given by Ada-DP-SPIDER satisfies: $\sigma \leq O\left(\sqrt{\frac{G^2 \log^2 d}{b_1} + \frac{M^2 \log^2 d}{b_2}\kappa}\right), r \leq O\left(\sqrt{\frac{G^2 \log(1/\delta)}{b_1^2 \epsilon^2} + \frac{M^2 \log(1/\delta)}{b_2^2 \epsilon^2}\kappa}\right)$.

The proof is given in Appendix C.1. To ensure that b_1 and b_2 remain valid mini-batch sizes under a fixed sample budget, we must control the number of times $\mathcal{O}_1$ is queried. Lemma 8 bounds the count:

Algorithm 2: Ada-DP-SPIDER

```
Input: DP budget $\epsilon$ and $\delta$, horizon $T$, model iterates $\{x_{t-1}\}_{t=1}^{T}$, drift threshold $\kappa$
$t \leftarrow 1$, $\mathsf{drift} \leftarrow \kappa$;
while $t \leq T$ do
    if $\mathit{drift} \geq \kappa$ then
        /* Using oracle $\mathcal{O}_1$ */
        Sample mini-batch $\mathcal{B}_t$ of size $b_1$ from $\mathcal{D}$;
        Sample $\xi_t \sim \mathcal{N}(0, c_1 \frac{G^2 \log \frac{1}{\delta}}{b_1^2 \epsilon^2} \mathbf{I}_d)$;
        $\hat{g}_t \leftarrow \mathcal{O}_1(x_{t-1}, \mathcal{B}_t) + \xi_t$;
        $\mathsf{drift} \leftarrow 0$;
    else
        /* Using oracle $\mathcal{O}_2$ */
        Sample mini-batch $\mathcal{B}_t$ of size $b_2$ from $\mathcal{D}$;
        Sample $\xi_t \sim \mathcal{N}(0,\ c_2 \frac{M^2 \log \frac{1}{\delta}}{b_2^2 \epsilon^2} \|x_{t-1} - x_{t-2}\|^2 \mathbf{I}_d)$;
        $\hat{g}_t \leftarrow \hat{g}_{t-1} + \mathcal{O}_2(x_{t-1}, x_{t-2}, \mathcal{B}_t) + \xi_t$;
    $\mathsf{drift} \leftarrow \mathsf{drift} + \eta^2 \|\hat{g}_t\|^2$;
    $t \leftarrow t + 1$;
Output: $\hat{g}_1, \hat{g}_2, \cdots, \hat{g}_T$
```

Lemma 8. Let Assumption 1 and 2 hold. Define $\mathcal{T} \coloneqq \{t \in [T] : \mathrm{drift}_t \geq \kappa\}$ as the set of rounds where the drift exceeds the threshold κ. With high probability (as in Theorem 1), $|\mathcal{T}| \leq O\left(U\eta/\kappa\right)$.

Proof is in Appendix C.2. Guided by Lemmas 7 and 8, we now derive the error bound for α via appropriate choices of b_1, b_2, and κ in Theorem 2. The proof is provided in Appendix C.3.

Theorem 2. Let Assumption 1 and 2 hold. Define $b_1 = \frac{n\kappa}{2U\eta}$, $b_2 = \frac{n\eta\chi^2}{2U}$ and $\kappa = \max\left\{\frac{G^{3/2}U^{1/2}\rho^{1/2}}{M^{5/2}n^{1/2}}, \frac{G^{14/15}d^{2/5}U^{4/5}\rho^{8/15}}{M^{34/15}(n\epsilon)^{4/5}}\right\}$. Then, running Gauss-PSGD with gradient oracle instantiated by Ada-DP-SPIDER ensures (ϵ, δ)-DP for constants c_1, c_2 and returns an α-SOSP with $\alpha = \tilde{O}\left(\frac{1}{n^{1/3}} + \left(\frac{\sqrt{d}}{n\epsilon}\right)^{2/5}\right)$.

Remark 2 (No Cyclic Dependency Among Parameters)**.** All algorithmic parameters are consistently defined in terms of the problem parameters n, d, and ϵ. Specifically, Gauss-PSGD parameters such as the step size η and the noise scale χ depend on the target error α (see (5)), and the gradient oracle parameters b_1 and b_2 are defined through η and χ, and thus also indirectly depend on α. In the proof of Theorem 2, by utilizing the relationship $\alpha = \tilde{O}(\sqrt{\sigma^2 + r^2 d})$, we obtain the closed-form expression of α that depends solely on the problem parameters n, d, and ϵ. As a result, all algorithm parameters are ultimately determined by n, d, and ϵ, and there is no cyclic dependency in the parameter design.

Algorithm 3: Distributed `Ada-DP-SPIDER`

```
Input: DP budget $\epsilon$ and $\delta$, horizon $T$, model iterates $\{x_{t-1}\}_{t=1}^{T}$, drift threshold $\kappa$
$t \leftarrow 1$, $\mathsf{drift} \leftarrow \kappa$;
while $t \leq T$ do
    if $\mathsf{drift} \geq \kappa$ then
        for every client $j$ in parallel do
            Sample mini-batch $\mathcal{B}_{j,t}$ of size $b_1$ from $\mathcal{D}_j$;
            Sample $\xi_{j,t} \sim \mathcal{N}(0, c_1 \frac{G^2 \log \frac{1}{\delta}}{b_1^2 \epsilon^2} \mathbf{I}_d)$;
            $\hat{g}_{j,t} \leftarrow \mathcal{O}_1(x_{t-1}, \mathcal{B}_{j,t}) + \xi_{j,t}$;
            Send $\hat{g}_{j,t}$ to the server;
        $\mathsf{drift} \leftarrow 0$;
    else
        for every client $i$ in parallel do
            Sample mini-batch $\mathcal{B}_{j,t}$ of size $b_2$ from $\mathcal{D}_j$;
            Sample $\xi_{j,t} \sim \mathcal{N}(0,\, c_2 \frac{M^2 \log \frac{1}{\delta}}{b_2^2 \epsilon^2} \|x_{t-1} - x_{t-2}\|^2 \mathbf{I}_d)$;
            $\hat{g}_{j,t} \leftarrow \hat{g}_{j,t-1} + \mathcal{O}_2(x_{t-1}, x_{t-2}, \mathcal{B}_{j,t}) + \xi_{j,t}$;
            Send $\hat{g}_{j,t}$ to the server;
    $\hat{g}_t \leftarrow \frac{1}{m} \sum_{j=1}^{m} \hat{g}_{j,t}$;
    $\mathsf{drift} \leftarrow \mathsf{drift} + \eta^2 \|\hat{g}_t\|^2$;
    $t \leftarrow t + 1$;
Output: $\hat{g}_1, \hat{g}_2, \cdots, \hat{g}_T$
```

7 Extension to Distributed SGD

By adapting the centralized gradient oracle `Ada-DP-SPIDER` (Algorithm 2) to the distributed setting, we obtain `Distributed Ada-DP-SPIDER` (Algorithm 3), enabling our `Gauss-PSGD` framework to extend seamlessly to distributed learning scenarios. The primary difference lies in the computation and communication scheme: in the distributed variant, each client performs local gradient estimation with private noise and communicates the privatized estimate to the server, which then aggregates the results. This avoids centralized access to raw data while still leveraging collective information.

The learning algorithm using `Distributed Ada-DP-SPIDER` can be viewed as an adaptive extension of the DIFF2 algorithm [36], which uses standard SPIDER and is limited to convergence to DP-FOSP under *homogeneous* data. To the best of our knowledge, our method is the first to achieve convergence to a DP-SOSP in a distributed setting with arbitrarily *heterogeneous* data.

Following the same analytical strategy as in Section 6, we first quantify in Lemma 9 the gradient estimation quality in the distributed case. The proof is provided in Appendix D.1.

Lemma 9. *Let Assumption 1 hold. For all $t \in [T]$, the distributed* `Ada-DP-SPIDER` *ensures that the gradient estimate $\hat{g}_t$ satisfies $\sigma \leq O\left(\sqrt{\frac{G^2 \log^2 d}{m \cdot b_1} + \frac{M^2 \log^2 d}{m \cdot b_2} \kappa}\right), r \leq O\left(\sqrt{\frac{G^2 \log \frac{1}{\delta}}{m \cdot b_1^2 \epsilon^2} + \frac{M^2 \log \frac{1}{\delta}}{m \cdot b_2^2 \epsilon^2} \kappa}\right)$.*

Algorithm 4: Private Model Selection in Distributed Learning

```
Input: Model iterates $\{x_t\}_{t=1}^{T}$, DP budget $\epsilon, \delta$
for $t \leftarrow 1, \cdots, T$ do
    for every client $j$ in parallel do
        Compute $\nabla \bar{F}_j(x_t) \leftarrow \nabla \hat{f}_{S_j}(x_t) + \theta_{i,t}$, where $\theta_{i,t} \sim \mathcal{N}\left(0, c_1 \frac{G^2 T \log(1/\delta)}{n^2\epsilon^2} \mathbf{I}_d\right)$ ;
        Compute $\nabla^2 \bar{F}_j(x_t) \leftarrow \nabla^2 \hat{f}_{S_j}(x_t) + \mathbf{H}_{j,t}$, where $\mathbf{H}_{j,t}$ is a symmetric matrix
          with its upper triangle (including the diagonal) being i.i.d. samples from
          $\mathcal{N}\left(0, c_2 \frac{M^2 d T \log(1/\delta)}{n^2\epsilon^2}\right)$ and each lower triangle entry is copied from its
          upper triangle counterpart;
        Send $\nabla \bar{F}_j(x_t)$ and $\nabla^2 \bar{F}_j(x_t)$ to the server;
    $\nabla \bar{F}(x_t) \leftarrow \frac{1}{m}\sum_{j=1}^{m} \nabla \bar{F}_j(x_t)$, $\nabla^2 \bar{F}(x_t) \leftarrow \frac{1}{m}\sum_{j=1}^{m} \nabla^2 \bar{F}_j(x_t)$;
    if $\|\nabla \bar{F}(x_t)\|_2 \le \alpha + \frac{G\log(8d/\omega')}{\sqrt{mn}} + \frac{G\sqrt{dT\log(1/\delta)\log(16/\omega')}}{\sqrt{m}n\epsilon}$ and
      $\lambda_{\min}\left(\nabla^2 \bar{F}(x_t)\right) \ge -\left(\sqrt{\rho\alpha} + M\sqrt{\frac{\log(8d/\omega')}{mn}} + \frac{Md\sqrt{T\log(1/\delta)\log(32/\omega')}}{\sqrt{m}n\epsilon}\right)$ then
        Return $x_t$
```

Based on this, we derive the error bound for α in the distributed setting. The proof is in Appendix D.2.

Theorem 3. Let Assumption 1 and 2 hold. Define $b_1 = \frac{n\kappa}{2U\eta}$, $b_2 = \frac{n\eta\chi^2}{2U}$ and $\kappa = \max\left\{\frac{G^{3/2}U^{1/2}\rho^{1/2}}{M^{5/2}(mn)^{1/2}}, \frac{G^{14/15}d^{2/5}U^{4/5}\rho^{8/15}}{M^{34/15}(\sqrt{m}n\epsilon)^{4/5}}\right\}$. Then, running `Gauss-PSGD` with gradient oracle instantiated by distributed `Ada-DP-SPIDER` ensures (ϵ, δ)-ICRL-DP for some constants c_1, c_2, and returns an α-SOSP with $\alpha = \tilde{O}\left(\frac{1}{(mn)^{1/3}} + \left(\frac{\sqrt{d}}{\sqrt{m}n\epsilon}\right)^{2/5}\right)$.

Remark 3. The error rate shown in Theorem 3 highlights the collaborative synergy among clients, indicating the learning performance benefits from distributed learning. Specifically, the first non-private term of α exhibits a linear dependence on m before n, while the second term, which accounts for the privacy cost, demonstrates a square root dependence $\sqrt{m}$ before n. This separation reflects the impact of data heterogeneity in distributed setting. The benefit of distributed collaboration under DP constraints is consistent with prior results in heterogeneous federated learning [15].

We conclude by demonstrating the advantages of our `Gauss-PSGD` framework in distributed learning by eliminating the need for a separate private model selection procedure. Without the guarantee of directly outputting an α-SOSP, one must resort to evaluating all model iterates generated during the learning process and privately selecting an approximate SOSP from them. As discussed in Appendix 3, the AboveThreshold mechanism used in [29] for the single-machine case is not applicable in distributed settings due to decentralized data access. To overcome this, we adapt [46, Algorithm 5] to the distributed setting, resulting in Algorithm 4. In this scheme, each client computes privatized gradients and Hessian estimates using additional local data, which are then aggregated by the server to evaluate the stationary point conditions. Suppose a distributed learning algorithm produces a sequence $\{x_t\}_{t\in[T]}$ that contains at least one α-DP-SOSP. The following result characterizes the quality of the point selected by Algorithm 4, whose proof is provided in Appendix D.3:

Theorem 4. Algorithm 4 satisfies (ϵ, δ)-ICRL-DP. Let Assumption 1 hold and $mn \geq \frac{4}{9}\log\frac{8d}{\omega'}$, then with probability at least $1 - \omega'$, if there exists an α-SOSP $x_p \in \{x_t\}_{t=1}^{T}$, then the selected point x_o is an α'-SOSP with $\alpha' = \tilde{O}\left(\alpha + \frac{1}{mn} + \frac{1}{\sqrt{mn}} + \frac{\alpha}{\sqrt{mn}} + \frac{\sqrt{d}}{\sqrt{m}n\epsilon\alpha^{5/4}} + \frac{d}{\sqrt{m}n\epsilon\alpha^{3/4}} + \frac{d^2}{mn^2\epsilon^2\alpha^{5/2}}\right)$.

Remark 4. To ensure that the selected model's error α' does not exceed the training error α, the following must hold: $\frac{\sqrt{d}}{\sqrt{m}n\epsilon\alpha^{5/4}} + \frac{d}{\sqrt{m}n\epsilon\alpha^{3/4}} + \frac{d^2}{mn^2\epsilon^2\alpha^{5/2}} \leq \tilde{O}(\alpha)$. This implies a constraint on the model dimension: $d \leq \min\{(\sqrt{m}n\epsilon)^2, (\sqrt{m}n\epsilon)^{6/13}\}$. Thus, in high-dimensional regimes, private model selection degrades the overall error rate, marking the limitation of selection-based approaches.

Remark 5. The error bound α' in Theorem 4 can be improved by estimating the smallest eigenvalue of the Hessian via Hessian-vector products using iterative methods such as the power method [25]. This reduces the dimensional dependence in the noise scale from $O(d)$ to $O(\sqrt{d})$. However, the remaining $\sqrt{d}$ factor is sill problematic in high-dimensional settings. In contrast, in the single-machine case, private model selection only requires perturbing scalar quantities, making the error independent of dimension, preserving the error guarantee of the learning algorithm. In distributed settings, sharing perturbed vectors becomes unavoidable. This emphasizes the necessity and superiority of our `Gauss-PSGD` framework that inherently avoids the need for any separate model selection step.

8 Limitation Discussion

One of the primary objective of this work is to rectify a key analytical error in [29] by presenting the correct error rates for DP stochastic non-convex optimization. Our proposed framework, `Gauss-PSGD`, is designed to be broadly applicable beyond the DP setting, offering a versatile optimization tool for general non-convex problems. Furthermore, this work makes the first attempt to extend DP-SOSP analysis to the distributed learning setting, establishing state-of-the-art utility guarantees.

To maintain consistency with prior work [29], we assume access to an unbiased gradient oracle. This assumption is fundamental in theoretical analysis and is also adopted by many recent studies in DP optimization and distributed learning, such as [1, 15]. However, it may not fully reflect the behavior of practical optimizers that employ biased and noisy gradients, particularly those using gradient clipping—a standard technique in DP implementations.

Nevertheless, our `Gauss-PSGD` framework can be extended to handle biased oracles induced by clipping. The main challenge lies in the analysis: incorporating clipping introduces bias, requiring a refined characterization of the descent dynamics. In particular, Lemma 3 (the descent lemma) must be adapted to reflect the bias–variance trade-off. Techniques for bias reduction in clipped DP learning—such as those developed in [53]—could offer a promising foundation for such an extension.

The saddle point escaping analysis (Lemma 1) can also be generalized. As shown in our proof, the key mechanism enabling escape is the injection of symmetric Gaussian noise, which drives the divergence in the coupling sequence. This mechanism remains valid under clipping, provided the Gaussian noise is appropriately calibrated. However, the number of steps required for escape may change due to the altered noise structure and bias, and a more delicate analysis would be required to quantify this behavior accurately.

We consider this as a promising direction for future work and leave its full exploration to subsequent studies.

9 Conclusion

In this work, we investigated the problem of finding second-order stationary points (SOSP) in differentially private (DP) stochastic non-convex optimization. We proposed a novel framework that leverages perturbed stochastic gradient descent (SGD) with Gaussian noise and introduces a novel criterion based on model drift distance to ensure provable saddle point escape and efficient convergence. By incorporating an adaptive SPIDER as the gradient oracle, we developed a new DP algorithm that rectifies existing error rates. Furthermore, we extended our approach to distributed learning scenarios with heterogeneous data, providing the first theoretical guarantees for finding DP-SOSP in such settings. Through rigorous analysis, we demonstrated that our framework not only avoids the pitfalls of private model selection but also remains effective in high-dimensional distributed learning environments.

Our work opens several promising directions for future research. A key challenge is bridging the gap between our upper bound and the existing DP lower bound for stochastic optimization, as established in [1]. The current lower bound is derived from convex loss functions and first-order stationary points, wheras finding DP-SOSP in non-convex optimization is inherently more difficult. We conjecture that the existing lower bound is not tight for the non-convex case. Establishing a tighter lower bound remains a critical open problem. Additionally, exploring whether our upper bounds can be further improved is another intriguing direction that warrants in-depth investigation.

References

[1] Raman Arora, Raef Bassily, Tomás González, Cristóbal A Guzmán, Michael Menart, and Enayat Ullah. Faster rates of convergence to stationary points in differentially private optimization. In *International Conference on Machine Learning*, pages 1060–1092. PMLR, 2023.

[2] Dmitrii Avdiukhin, Michael Dinitz, Chenglin Fan, and Grigory Yaroslavtsev. Noise is all you need: Private second-order convergence of noisy sgd. *arXiv preprint arXiv:2410.06878*, 2024.

[3] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. *Advances in neural information processing systems*, 32, 2019.

[4] Raef Bassily, Cristóbal Guzmán, and Michael Menart. Differentially private stochastic optimization: New results in convex and non-convex settings. *Advances in Neural Information Processing Systems*, 34:9317–9329, 2021.

[5] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

[6] Sijin Chen, Zhize Li, and Yuejie Chi. Escaping saddle points in heterogeneous federated learning via distributed sgd with communication compression. In *International Conference on Artificial Intelligence and Statistics*, pages 2701–2709. PMLR, 2024.

[7] Christopher A Choquette-Choo, Arun Ganesh, and Abhradeep Thakurta. Optimal rates for dp-sco with a single epoch and large batches. *arXiv preprint arXiv:2406.02716*, 2024.

[8] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

[9] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In *International Conference on Machine Learning*, pages 1155–1164. PMLR, 2018.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[11] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[12] 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.

[13] Cong Fang, Zhouchen Lin, and Tong Zhang. Sharp analysis for nonconvex sgd escaping from saddle points. In *Conference on Learning Theory*, pages 1192–1234. PMLR, 2019.

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

[15] Changyu Gao, Andrew Lowy, Xingyu Zhou, and Stephen J Wright. Private heterogeneous federated learning without a trusted server revisited: Error-optimal and communication-efficient algorithms for convex losses. *arXiv preprint arXiv:2407.09690*, 2024.

[16] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.

[17] Rong Ge, Zhize Li, Weiyao Wang, and Xiang Wang. Stabilized svrg: Simple variance reduction for nonconvex optimization. In *Conference on learning theory*, pages 1394–1448. PMLR, 2019.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[19] Lijie Hu, Shuo Ni, Hanshen Xiao, and Di Wang. High dimensional differentially private stochastic optimization with heavy-tailed data. In *Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 227–236, 2022.

[20] Prateek Jain, Chi Jin, Sham M Kakade, and Praneeth Netrapalli. Computing matrix squareroot via non convex local search. *arXiv preprint arXiv:1507.05854*, 2015.

[21] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pages 1724–1732. PMLR, 2017.

[22] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*, 2019.

[23] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29, 2021.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[25] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. 1950.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[27] Zhize Li. Ssrgd: Simple stochastic recursive gradient descent for escaping saddle points. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] Daogao Liu and Hilal Asi. User-level differentially private stochastic convex optimization: Efficient algorithms with optimal rates. In *International Conference on Artificial Intelligence and Statistics*, pages 4240–4248. PMLR, 2024.

[29] Daogao Liu, Arun Ganesh, Sewoong Oh, and Abhradeep Guha Thakurta. Private (stochastic) non-convex optimization revisited: Second-order stationary points and excess risks. *Advances in Neural Information Processing Systems*, 36, 2024.

[30] Ruixuan Liu, Yang Cao, Hong Chen, Ruoyang Guo, and Masatoshi Yoshikawa. Flame: Differentially private federated learning in the shuffle model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8688–8696, 2021.

[31] Andrew Lowy, Ali Ghafelebashi, and Meisam Razaviyayn. Private non-convex federated learning without a trusted server. In *International Conference on Artificial Intelligence and Statistics*, pages 5749–5786. PMLR, 2023.

[32] Andrew Lowy and Meisam Razaviyayn. Private federated learning without a trusted server: Optimal algorithms for convex losses. In *The Eleventh International Conference on Learning Representations*, 2023.

[33] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[34] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 28th ACM SIGMOD International Conference on Management of data (SIGMOD)*, pages 19–30, 2009.

[35] Tomoya Murata and Taiji Suzuki. Escaping saddle points with bias-variance reduced local perturbed sgd for communication efficient nonconvex distributed learning. *Advances in Neural Information Processing Systems*, 35:5039–5051, 2022.

[36] Tomoya Murata and Taiji Suzuki. Diff2: Differential private optimization via gradient differences for nonconvex distributed learning. In *International Conference on Machine Learning*, pages 25523–25548. PMLR, 2023.

[37] 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.

[38] Jinyan Su, Lijie Hu, and Di Wang. Faster rates of private stochastic convex optimization. In *International Conference on Algorithmic Learning Theory*, pages 995–1002. PMLR, 2022.

[39] Jinyan Su, Lijie Hu, and Di Wang. Faster rates of differentially private stochastic convex optimization. *Journal of Machine Learning Research*, 25(114):1–41, 2024.

[40] Jinyan Su, Changhong Zhao, and Di Wang. Differentially private stochastic convex optimization in (non)-euclidean space revisited. In *Uncertainty in Artificial Intelligence*, pages 2026–2035. PMLR, 2023.

[41] Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 2379–2383. IEEE, 2016.

[42] Youming Tao, Shuzhen Chen, Congwei Zhang, Di Wang, Dongxiao Yu, Xiuzhen Cheng, and Falko Dressler. Private over-the-air federated learning at band-limited edge. *IEEE Transactions on Mobile Computing*, 2024.

[43] Youming Tao, Yulian Wu, Xiuzhen Cheng, and Di Wang. Private stochastic convex optimization and sparse learning with heavy-tailed data revisited. In *IJCAI*, pages 3947–3953, 2022.

[44] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12:389–434, 2012.

[45] Roman Vershynin. High-dimensional probability. *University of California, Irvine*, 10:11, 2020.

[46] Di Wang, Changyou Chen, and Jinhui Xu. Differentially private empirical risk minimization with non-convex loss functions. In *International Conference on Machine Learning*, pages 6526–6535. PMLR, 2019.

[47] Di Wang, Marco Gaboardi, Adam Smith, and Jinhui Xu. Empirical risk minimization in the non-interactive local model of differential privacy. *Journal of machine learning research*, 21(200):1–39, 2020.

[48] Di Wang, Marco Gaboardi, and Jinhui Xu. Empirical risk minimization in non-interactive local differential privacy revisited. *Advances in Neural Information Processing Systems*, 31, 2018.

[49] Di Wang, Hanshen Xiao, Srinivas Devadas, and Jinhui Xu. On differentially private stochastic convex optimization with heavy-tailed data. In *International Conference on Machine Learning*, pages 10081–10091. PMLR, 2020.

[50] Di Wang and Jinhui Xu. Escaping saddle points of empirical risk privately and scalably via dp-trust region method. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*, pages 90–106. Springer, 2021.

[51] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

[52] Zihang Xiang, Tianhao Wang, Wanyu Lin, and Di Wang. Practical differentially private and byzantine-resilient federated learning. *Proceedings of the ACM on Management of Data*, 1(2):1–26, 2023.

[53] Hanshen Xiao, Zihang Xiang, Di Wang, and Srinivas Devadas. A theory to instruct differentially-private learning via clipping bias reduction. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2170–2189. IEEE, 2023.

[54] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Defending against saddle point attack in byzantine-robust distributed learning. In *International Conference on Machine Learning*, pages 7074–7084. PMLR, 2019.

[55] Ruijia Zhang, Mingxi Lei, Meng Ding, Zihang Xiang, Jinhui Xu, and Di Wang. Improved rates of differentially private nonconvex-strongly-concave minimax optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22524–22532, 2025.

[56] Yingxue Zhou, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Arindam Banerjee. Private stochastic non-convex optimization: Adaptive algorithms and tighter generalization bounds. *arXiv preprint arXiv:2006.13501*, 2020.

A Useful Facts for Analysis

A.1 Probability Tools

Definition 6 (Sub-Gaussian random vector [22, Definition 2])**.** A random vector $v \in \mathbb{R}^d$ is ζ*-sub-Gaussian* (or SG(ζ)), if there exists a positive constant ζ such that

$$\mathbb{E}[\exp(\langle u, v - \mathbb{E}[v] \rangle)] \leq \exp\left(\frac{\|u\|_2^2 \zeta^2}{2}\right), \qquad \forall u \in \mathbb{R}^d. \tag{10}$$

Definition 7 (Norm-sub-Gaussian random vector [22, Definition 3])**.** A random vector $v \in \mathbb{R}^d$ is ζ*-norm-sub-Gaussian* (or nSG(ζ)), if there exists a positive constant ζ such that

$$\mathbb{P}\left[\|v - \mathbb{E}[v]\| \geq t\right] \leq 2\exp\left(-\frac{t^2}{2\zeta^2}\right), \qquad \forall t \in \mathbb{R}. \tag{11}$$

Note that norm-sub-Gaussian random vectors (Definition 7) are more general than sub-Gaussian random vectors (Definition 6), as sub-Gaussian distributions require *isotropy*, whereas norm-sub-Gaussian distributions do not impose this condition.

Lemma 10 ([22, Lemma 1])**.** A SG(r) random vector $v \in \mathbb{R}^d$ is also nSG($2\sqrt{2} \cdot r\sqrt{d}$).

We are interested in the properties of norm-subGaussian martingale difference sequences. Concretely, they are sequences satisfying the following properties.

Condition 1. Consider random vectors $v_1, \cdots, v_p \in \mathbb{R}^d$, and corresponding filtrations $\mathcal{F}_i = \sigma(v_1, \cdots, v_i)$ for $i \in [n]$, such that $v_i | \mathcal{F}_{i-1}$ is zero-mean nSG(ζ_i) with $\zeta_i \in \mathcal{F}_{i-1}$. That is,

$$\mathbb{E}[v_i | \mathcal{F}_{i-1}] = 0, \qquad \mathbb{P}\left[\|v_i\| \geq t | \mathcal{F}_{i-1}\right] \leq 2\exp\left(-\frac{t^2}{2\zeta^2}\right), \qquad \forall t \in \mathbb{R}, \forall i \in [p]. \tag{12}$$

Lemma 11 (Hoeffding type inequality for norm-sub-Gaussian [22, Corollary 7])**.** Let random vectors $v_1, \cdots, v_p \in \mathbb{R}^d$, and corresponding filtrations $\mathcal{F}_i = \sigma(v_1, \cdots, v_i)$ for $i \in [k]$ satisfy condition 1 with fixed $\{\zeta_i\}$. Then for any $\iota > 0$, there exists an absolute constant C such that, with probability at least $1 - 2d \cdot e^{-\iota}$,

$$\left\|\sum_{i=1}^{p} v_i\right\|_2 \leq C \cdot \sqrt{\sum_{i=1}^{p} \zeta_i^2 \cdot \iota}. \tag{13}$$

Lemma 11 implies that the sum of norm-sub-Gaussian random vectors is till norm-sub-Gaussian.

Corollary 3. Let random vectors $v_1, \cdots, v_p \in \mathbb{R}^d$, and corresponding filtrations $\mathcal{F}_i = \sigma(v_1, \cdots, v_i)$ for $i \in [k]$ satisfy condition 1 with fixed $\{\zeta_i\}$. Then $\sum_{i=1}^{p} v_i$ is nSG $\left(C \cdot \sqrt{\log(d) \sum_{i=1}^{k} \zeta_i^2}\right)$.

Proof. Let $\zeta_+ := \sqrt{C \log(d) \sum_{i=1}^{k} \zeta_i}$. According to Definition 7, we aim to show that, for any $\omega \in (0, 1)$, with probability at least $1 - \omega$, $\|\sum_{i=1}^{p} v_i\| \leq \sqrt{2\zeta_+^2 \ln \frac{2}{\omega}}$. By Lemma 11, we have known that, with probability at least $1 - \omega$, $\|\sum_{i=1}^{p} v_i\| \leq C \cdot \sqrt{\sum_{i=1}^{p} \zeta_i^2 \ln \frac{2d}{\omega}}$.

Next, we show that $\sqrt{2\zeta_+^2 \ln\frac{2}{\omega}} \geq C \cdot \sqrt{\sum_{i=1}^{p} \zeta_i^2 \ln\frac{2d}{\omega}}$, which, by re-arranging the terms, is equivalent to show $\zeta_+^2 \geq \frac{C^2}{2}(\sum_{i=i}^{p} \zeta_i^2)\frac{\log\frac{2d}{\omega}}{\log\frac{2}{\omega}}$. This follows directly from the fact that $\frac{\log\frac{2d}{\omega}}{\log\frac{2}{\omega}} \leq 2\log d, \forall \omega \in (0,1)$. □

Lemma 12 ([23, Lemma C.6])**.** Let random vectors $v_1, \cdots, v_p \in \mathbb{R}^d$, and corresponding filtrations $\mathcal{F}_i = \sigma(v_1, \cdots, v_i)$ for $i \in [k]$ satisfy condition 1, then for any $\iota > 0$, and $B > b > 0$, there exists an absolute constant C such that, with probability at least $1 - 2d\log\left(\frac{B}{b}\right) \cdot e^{-\iota}$,

$$\sum_{i=1}^{p} \zeta_i^2 \geq B \qquad \text{or} \qquad \left\|\sum_{i=i}^{p} v_i\right\| \leq C \cdot \sqrt{\max\left\{\sum_{i}^{p} \zeta_i^2, b\right\} \cdot \iota}. \tag{14}$$

Lemma 13 ([23, Lemma C.7])**.** Let random vectors $v_1, \cdots, v_p \in \mathbb{R}^d$, and corresponding filtrations $\mathcal{F}_i = \sigma(v_1, \cdots, v_i)$ for $i \in [k]$ satisfy condition 1 with fixed $\zeta_1 = \zeta_2 = \cdots = \zeta_p = \zeta$, then there exists an absolute constant C such that, for any $\iota > 0$, with probability at least $1 - e^{-\iota}$,

$$\sum_{i=1}^{p} \|v_i\|^2 \leq C \cdot \zeta^2 \cdot (p + \iota). \tag{15}$$

Lemma 14 (Matrix Bernstein inequality [44, Theorem 1.4])**.** Consider a finite sequence $\{\mathbf{M}_i\}_{i\in[k]}$ of independent, random, self-adjoint matrices with dimension $d \times d$. Assume that each random matrix satisfies $\mathbb{E}[\mathbf{M}_i] = \mathbf{0}$, $\|\mathbf{M}_i\|_2 \leq B$, then for all $t \geq 0$, we have

$$\mathbb{P}\left[\left\|\sum_{i\in[k]} \mathbf{M}_i\right\|_2 \geq t\right] \leq d\exp\left(-\frac{t^2}{2(\sigma^2 + Bt/3)}\right), \tag{16}$$

where $\sigma^2 = \left\|\sum_{i\in[k]} \mathbb{E}[\mathbf{M}_i^2]\right\|_2$.

Lemma 15 (Norm of symmetric matrices with sub-gaussian entries [45, Corollary 4.4.8])**.** Let $\mathbf{M}$ be an $d \times d$ symmetric random matrix whose entries $\mathbf{M}_{i,j}$ on and above the diagonal are independent, mean zero, sub-gaussian random variables. Then, with probability at least $1 - 4\exp(-t^2)$, for any $t > 0$ we have

$$\|\mathbf{M}\|_2 \leq C \cdot \max_{i,j} \|\mathbf{M}_{i,j}\|_{\psi_2} \cdot (\sqrt{d} + t), \tag{17}$$

where C is a universal constant.

A.2 Privacy Preliminaries

Definition 8 (Gaussian Mechanism [11])**.** Given any input data $D \in \mathcal{X}^n$ and a query function $q : \mathcal{X}^n \to \mathbb{R}^d$, the Gaussian mechanism $\mathcal{M}_G$ is defined as $q(D) + \nu$ where $\nu \sim \mathcal{N}(0, \sigma_G^2 \mathbf{I}_d)$. Let $\Delta_2(q)$ be the ℓ_2-sensitivity of q, *i.e.*, $\Delta_2(q) := \sup_{D\sim D'} \|q(D) - q(D')\|_2$. For any $\sigma, \delta > 0$, $\mathcal{M}_G$ guarantees $(\frac{\Delta_2(q)}{\sigma_G}\sqrt{2\log\frac{1.25}{\delta}}, \delta)$-DP. That is, if we want the output of q to be (ϵ, δ)-DP for any $0 < \epsilon, \delta < 1$, then σ_G should be set to $\frac{\Delta_2(q)}{\epsilon}\sqrt{2\log\frac{1.25}{\delta}}$.

Lemma 16 (Adaptive Composition Theorem [11])**.** Given target privacy parameters $0 < \epsilon < 1$ and $0 < \delta < 1$, to ensure (ϵ, δ)-DP over k-fold adaptive mechanisms, it suffices that each mechanism is (ϵ', δ')-DP, where $\epsilon' = \frac{\epsilon}{2\sqrt{2k\ln(2/\delta)}}$ and $\delta' = \frac{\delta}{2k}$.

Lemma 17 (Parallel Composition of DP [34])**.** Suppose there are n (ϵ, δ)-differentially private mechanisms $\{\mathcal{M}_i\}_{i=1}^n$ and n disjoint datasets denoted by $\{D_i\}_{i=1}^n$. Then the algorithm, which applies each $\mathcal{M}_i$ on the corresponding D_i, preserves (ϵ, δ)-DP in total.

B Omitted Proofs in Section 5

B.1 Proof of Lemma 1

Proof of Lemma 1. We begin by introducing the following notations:

$$\hat{x}_t \coloneqq x_t - x'_t, \tag{18}$$

$$\hat{\zeta}_t \coloneqq \zeta_t - \zeta'_t, \tag{19}$$

$$\hat{\xi}_t \coloneqq \xi_t - \xi'_t, \tag{20}$$

$$\Delta_t \coloneqq \int_0^1 \nabla^2 F(y \cdot x_t + (1-y) \cdot x'_t) \, \mathrm{d}y - \mathcal{H} \tag{21}$$

The proof strategy is to derive a contradiction by showing that if the model remains localized (i.e., stays within a radius $\mathcal{R}$ around the saddle point) with high probability, then the coupling sequence must still diverge with non-negligible probability.

We first characterize the dynamics of $\hat{x}_t$ in the following Lemma 18. At a high level, we decompose the difference of the coupling sequence x_t into three components: **(i)** a curvature-dependent term $\mathscr{P}_h(t)$, **(ii)** a stochastic gradient noise term $\mathscr{P}_{sg}(t)$, **(iii)** a perturbation-driven term $\mathscr{P}_p(t)$.

Lemma 18 (Coupling Dynamics)**.** For any $t \geq 0$, the difference between the two coupled iterates satisfies:

$$\hat{x}_t = \underbrace{-\eta \sum_{i=1}^{t} (\mathbf{I}_d - \eta\mathcal{H})^{t-i} \Delta_{i-1} \hat{x}_{i-1}}_{\mathscr{P}_h(t)} \underbrace{- \eta \sum_{i=1}^{t} (\mathbf{I}_d - \eta\mathcal{H})^{t-i} \hat{\zeta}_i}_{\mathscr{P}_{sg}(t)} \underbrace{- \eta \sum_{i=1}^{t} (\mathbf{I}_d - \eta\mathcal{H})^{t-i} \hat{\xi}_i}_{\mathscr{P}_p(t)}. \tag{22}$$

Proof of Lemma 18. By the update rule:

$$\begin{aligned} \hat{x}_t &= x_t - x'_t & (23)\\ &= \hat{x}_{t-1} - \eta[\nabla F(x_{t-1}) - \nabla F(x'_{t-1}) + \zeta_t - \zeta'_t + \xi_t - \xi'_t] & (24)\\ &= \hat{x}_{t-1} - \eta[(\mathcal{H} + \Delta_{t-1})\hat{x}_{t-1} + \hat{\zeta}_t + \hat{\xi}_t] & (25)\\ &= (\mathbf{I}_d - \eta\mathcal{H})\hat{x}_{t-1} - \eta[\Delta_{t-1}\hat{x}_{t-1} + \hat{\zeta}_t + \hat{\xi}_t]. & (26) \end{aligned}$$

Unrolling the recursion with initial condition $\hat{x}_0 = 0$ yields the desired result:

$$\begin{aligned} \hat{x}_t &= (\mathbf{I}_d - \eta\mathcal{H})^t \hat{x}_0 - \eta \sum_{i=1}^{t} (\mathbf{I}_d - \eta\mathcal{H})^{t-i} (\Delta_{i-1}\hat{x}_{i-1} + \hat{\zeta}_i + \hat{\xi}_i) & (27)\\ &= -\eta \sum_{i=1}^{t} (\mathbf{I}_d - \eta\mathcal{H})^{t-i} (\Delta_{i-1}\hat{x}_{i-1} + \hat{\zeta}_i + \hat{\xi}_i). & (28) \end{aligned}$$

□

Let $\mathcal{E}$ denote the event that both sequences remain localized:

$$\mathcal{E} := \left\{ \forall t \leq \Gamma : \max\left\{ \|x_t - \tilde{x}\|, \|x'_t - \tilde{x}\| \right\} \leq \mathcal{R} \right\}.$$

We proceed by contradiction. Assume:

$$\mathbb{P}(\mathcal{E}) \geq \frac{3}{4}. \tag{29}$$

To derive a contradiction, we analyze the terms in (22), showing in Lemma 19 and Lemma 20 that the perturbation term $\mathscr{P}_p(t)$ dominates, while the curvature and stochastic gradient terms remain controlled. Define:

$$\mathfrak{a}(t) := \sqrt{\sum_{i=1}^{t} (1+\eta\gamma)^{2(t-i)}}, \qquad \mathfrak{b}(t) := \frac{(1+\eta\gamma)^t}{\sqrt{2\eta\gamma}}. \tag{30}$$

It has been verified in [23, Lemma 29] that $\mathfrak{a}(t) \leq \mathfrak{b}(t)$ for all $t \in \mathbb{N}$.

Lemma 19. For all $t \geq 0$, the following hold:

$$\mathbb{P}\left[\|\mathscr{P}_p(t)\| \leq c\mathfrak{b}(t)\eta r \cdot \sqrt{\iota} \right] \geq 1 - 2e^{-\iota} \tag{31}$$

$$\mathbb{P}\left[\|\mathscr{P}_p(t)\| \geq \frac{\mathfrak{b}(\Gamma)\eta r}{10} \right] \geq \frac{2}{3} \tag{32}$$

The proof follows from standard Gaussian concentration and is omitted here; see [23, Lemma 30].

Lemma 20. For all $t \geq 0$, conditioned on $\mathcal{E}$, we have:

$$\mathbb{P}\left[\|\mathscr{P}_h(t) + \mathscr{P}_{sg}(t)\| \leq \frac{\mathfrak{b}(t)\eta r}{20} \middle| \mathcal{E} \right] \geq 1 - 6d\Gamma \log\left(\frac{\mathcal{R}}{\eta r}\right) e^{-\iota} \tag{33}$$

Proof of Lemma 20. We prove the following strengthened claim for any $t \leq \Gamma$ by induction:

$$\mathbb{P}\left[\forall i \leq t : \|\mathscr{P}_h(i) + \mathscr{P}_{sg}(i)\| \leq \frac{\mathfrak{b}(i)\eta r}{20}, \|\mathscr{P}_p(i)\| \leq c\mathfrak{b}(i)\eta r\sqrt{\iota} \middle| \mathcal{E} \right] \leq 1 - 6dt \log\left(\frac{\mathcal{R}}{\eta r}\right) e^{-\iota}. \tag{34}$$

For the base case of $t = 0$, the claim holds trivially as $\mathscr{P}_h(0) = \mathscr{P}_{sg}(0) = 0$. Suppose the claim holds for a step $t < \Gamma$, we then forward prove that the claim also holds for step $t+1 \leq \Gamma$. Since for $\forall i \leq t$, $\|\mathscr{P}_p(i)\| \leq c\mathfrak{b}(i)\eta r\sqrt{\iota}$, we have

$$\begin{aligned} \|\hat{x}_i\| &\leq \|\mathscr{P}_h(i) + \mathscr{P}_{sg}(i)\| + \|\mathscr{P}_p(i)\| && (35) \\ &\leq \frac{\mathfrak{b}(i)\eta r}{20} + c\mathfrak{b}(i)\eta r \cdot \sqrt{\iota} && (36) \\ &\leq 2c\mathfrak{b}(i)\eta r \cdot \sqrt{\iota}. && (37) \end{aligned}$$

Moreover, due to assumption (29) and the Hessian Lipschitz property, we have

$$\begin{aligned} \|\Delta_i\| &= \int_0^1 \nabla^2 F(y \cdot x_i + (1-y) \cdot x'_i) \, \mathrm{d}y && (38) \\ &\leq \rho \max\{\|x_i - \tilde{x}\|, \|x'_i - \tilde{x}\|\} \leq \rho\mathcal{R}. && (39) \end{aligned}$$

With the above upper bounds on $\|\hat{x}_i\|$ and $\|\Delta_i\|$ for $i \le t$, we immediately get for case $t+1$ from the definition of $\mathscr{P}_h(\cdot)$ in (22) that

$$\|\mathscr{P}_h(t+1)\| \le \eta\rho\mathcal{R}\sum_{i=1}^{t+1}(1+\eta\gamma)^{t+1-i}\left(2c\mathfrak{b}(i)\eta r\sqrt{\iota}\right) \tag{40}$$

$$\le 2\eta\rho\mathcal{R}\Gamma c\mathfrak{b}(t+1)\eta r\sqrt{\iota} \le \frac{\mathfrak{b}(t+1)\eta r}{40}, \tag{41}$$

where the last inequality follows from $2c\eta\rho\mathcal{R}\Gamma = \frac{2c}{s} \le \frac{1}{40}$ for large enough s such that $s \ge 80c$.

Note that $\hat{\zeta}_t|\mathcal{F}_{t-1} \sim \mathrm{nSG}(M\|\hat{x}_t\|)$, by applying Lemma 12 with $B = [\mathfrak{a}(t)]^2\eta^2M^2\mathcal{R}^2$ and $b = [\mathfrak{a}(t)]^2\eta^2M^2\eta^2r^2$ therein, we know that, with probability at least $1 - 4d\log\left(\frac{\mathcal{R}}{\eta r}\right)e^{-\iota}$, we have

$$\|\mathscr{P}_{sg}(t+1)\| \le 2c\eta M\sqrt{\Gamma}\mathfrak{b}(t)\eta r\sqrt{\iota}. \tag{42}$$

For large enough s such that $s \ge (80c)^2$, we have $c\eta M\sqrt{\Gamma\iota} \le \frac{2c}{\sqrt{s}} \le \frac{1}{40}$. Thus,

$$\|\mathscr{P}_{sg}(t+1)\| \le c\eta M\sqrt{\Gamma}\mathfrak{b}(t)\eta r\sqrt{\iota} \le \frac{\mathfrak{b}(t)\eta r}{40}. \tag{43}$$

By Lemma 19, we know that, for case $t+1$, with probability at least $1 - 2e^{-\iota}$, we have

$$\|\mathscr{P}_p(t+1)\| \le c\mathfrak{b}(t+1)\eta r\sqrt{\iota} \tag{44}$$

By the union bound, with probability at least $1 - \left(6dt\log\left(\frac{\mathcal{R}}{\eta r}\right)e^{-\iota} + 4d\log\left(\frac{\mathcal{R}}{\eta r}\right)e^{-\iota} + 2e^{-\iota}\right) \ge 1 - 6d(t+1)\log\left(\frac{\mathcal{R}}{\eta r}\right)e^{-\iota}$,

$$\|\mathscr{P}_h(t+1) + \mathscr{P}_{sg}(t+1)\| \le \frac{\mathfrak{b}(t)\eta r}{20} \le \frac{\mathfrak{b}(t+1)\eta r}{20}, \qquad \|\mathscr{P}_p(t+1)\| \le c\mathfrak{b}(t+1)\eta r\sqrt{\iota}, \tag{45}$$

which concludes the proof. □

Now we complete the proof of Lemma 1. Choose ι large enough such that

$$\iota \ge \log\left(36d\Gamma\log\left(\frac{\mathcal{R}}{\eta r}\right)\right), \tag{46}$$

which is promised by $\mu \ge \frac{1}{s}\log\left(\frac{9d}{C^{\frac{1}{4}}\eta\sqrt{s\rho\psi}}\log\left(\frac{4C^{\frac{1}{4}}}{s\eta r}\sqrt{\frac{\psi}{\rho}}\right)\right)$ for sufficiently large numerical constant s. Then we have:

$$6d\Gamma\log\left(\frac{\mathcal{R}}{\eta r}\right)e^{-\iota} \le \frac{2}{9}. \tag{47}$$

From Lemma 19, we have:

$$\mathbb{P}\left[\|\mathscr{P}_p(\Gamma)\| \ge \frac{\mathfrak{b}(\Gamma)\eta r}{10}\right] \ge \frac{2}{3}, \tag{48}$$

and from Lemma 20,

$$\mathbb{P}\left[\|\mathscr{P}_h(\Gamma) + \mathscr{P}_{sg}(\Gamma)\| \le \frac{\mathfrak{b}(\Gamma)\eta r}{20}\right] \ge \frac{3}{4}\cdot\left(1 - 6d\Gamma\log\left(\frac{\mathcal{R}}{\eta r}\right)e^{-\iota}\right) \ge \frac{7}{12} \tag{49}$$

By the union bound, with probability at least $1 - \left(1 - \frac{2}{3}\right) - \left(1 - \frac{7}{12}\right) = \frac{1}{4}$, both events hold:

$$\|\mathscr{P}_p(\Gamma)\| \geq \frac{\mathfrak{b}(\Gamma)\eta r}{10}, \quad \|\mathscr{P}_h(\Gamma) + \mathscr{P}_{sg}(\Gamma)\| \leq \frac{\mathfrak{b}(\Gamma)\eta r}{20}. \tag{50}$$

Therefore, using the triangle inequality:

$$\max\left\{\|x_\Gamma - \tilde{x}\|, \|x'_\Gamma - \tilde{x}\|\right\} \tag{51}$$

$$\geq \frac{1}{2}\|\hat{x}_\Gamma\| \geq \frac{1}{2}\left[\|\mathscr{P}_p(\Gamma)\| - \|\mathscr{P}_h(\Gamma) + \mathscr{P}_{sg}(\Gamma)\|\right] \geq \frac{\mathfrak{b}(\Gamma)\eta r}{40} = \frac{(1+\eta\gamma)^\Gamma \sqrt{\eta r}}{40\sqrt{2}} \tag{52}$$

$$\geq \frac{(1+\eta\sqrt{\rho\alpha})^\Gamma \sqrt{\eta r}}{40\sqrt{2}} \geq \frac{2^{\eta\sqrt{\rho\alpha}\Gamma}\sqrt{\eta r}}{40\sqrt{2}} = \frac{2^{\frac{\iota}{s}}\sqrt{\eta r}}{40\sqrt{2}} = \frac{2^{\mu}\sqrt{\eta r}}{40\sqrt{2}} > \mathcal{R}, \tag{53}$$

where the second last inequality is due to the fact $1+a > 2^a, \forall a \in (0,1]$ and $\eta\sqrt{\rho\alpha} \leq \frac{1}{\iota^2} \leq 1$, and the last inequality is because $\mu > \log\left(\frac{160\sqrt{2}C^{\frac{1}{4}}}{s\sqrt{\eta r}}\sqrt{\frac{\psi}{\rho}}\right)$.

The above means that the localization event $\mathcal{E}$ fails with probability at least 1/4, i.e., $\mathbb{P}(\mathcal{E}) < \frac{3}{4}$, which contradicts with our assumption (29). Therefore, the assumption (29) should be false, that is, with probability at least $\frac{1}{4}$, $\exists t \leq \Gamma, \max\{\|x_t - \tilde{x}\|, \|x'_t - \tilde{x}\|\} \geq \mathcal{R}$, completing the proof. □

B.2 Proof of Lemma 2

Proof of Lemma 2. The failure probability after Q independent repetitions is at most $(7/8)^Q$. Setting $Q = \frac{26}{5}\log(1/\omega_0)$ yields $(7/8)^Q \leq \omega_0$, completing the proof. □

B.3 Proof of Lemma 3

Proof of Lemma 3. For any $t \geq 1$, by M-smoothness of F, we have:

$$\begin{aligned}
F(x_t) - F(x_{t-1}) &\leq \langle \nabla F(x_{t-1}), x_t - x_{t-1}\rangle + \frac{M}{2}\|x_t - x_{t-1}\|^2 && (54)\\
&\leq -\eta\langle \nabla F(x_{t-1}), \hat{g}_{t-1}\rangle + \frac{M}{2}\eta^2\|\hat{g}_{t-1}\|^2 && (55)\\
&\leq -\eta\langle \nabla F(x_{t-1}), \hat{g}_{t-1}\rangle + \frac{\eta}{2}\|\hat{g}_{t-1}\|^2 && (56)\\
&\leq \frac{\eta}{2}\|\nu_t\|^2 - \frac{\eta}{2}\|\nabla F(x_{t-1})\|^2 - \frac{\eta}{2}\|\hat{g}_{t-1}\|^2 + \frac{\eta}{2}\|\hat{g}_{t-1}\|^2 && (57)\\
&= -\frac{\eta}{2}\|\nabla F(x_{t-1})\|^2 + \frac{\eta}{2}\|\nu_t\|^2. && (58)
\end{aligned}$$

Summing from t_0+1 to t_0+t, we obtain:

$$F(x_{t_0+t}) - F(x_{t_0}) \leq -\frac{\eta}{2}\sum_{i=0}^{t-1}\|\nabla F(x_{t_0+i})\|^2 + \frac{\eta}{2}\sum_{i=1}^{t}\|\nu_{t_0+i}\|^2 \tag{59}$$

□

B.4 Proof of Corollary 2

Proof of Corollary 2. Note that

$$\frac{\eta}{2}\sum_{i=1}^{t}\|\nu_{t_0+i}\|^2 = \frac{\eta}{2}\sum_{i=1}^{t}\|\zeta_{t_0+i}+\xi_{t_0+i}\|^2 \leq \eta\sum_{i=1}^{t}(\|\zeta_{t_0+i}\|^2+\|\xi_{t_0+i}\|^2) \tag{60}$$

By Lemma 13, since $\zeta_i \sim \mathrm{nSG}(\sigma)$, with probability at least $1-e^{-\iota}$:

$$\sum_{i=1}^{t}\|\zeta_{t_0+i}\|^2 \leq C\cdot\sigma^2(t+\iota). \tag{61}$$

Using Lemma 10, each $\xi_i \sim \mathrm{nSG}(2\sqrt{2}r\sqrt{d})$, and applying Lemma 13 again, with probability at least $1-e^{-\iota}$:

$$\sum_{i=1}^{t}\|\xi_{t_0+i}\|^2 \leq 8C\cdot r^2 d(t+\iota). \tag{62}$$

By the union bound, both bounds hold with probability at least $1-2e^{-\iota}$. □

B.5 Proof of Lemma 4

Proof of Lemma 4. We begin with:

$$\|x_{t_0+\tau}-x_{t_0}\|^2 = \eta^2\left\|\sum_{t=1}^{\tau}\nabla F(x_{t_0+t-1})+\nu_{t_0+t}\right\|^2 \tag{63}$$

$$\leq 2\eta^2\tau\sum_{t=1}^{\tau}\left(\|\nabla F(x_{t_0+t-1})\|^2+\|\nu_{t_0+t}\|^2\right). \tag{64}$$

Following the same argument in the proof of corollary 2, with probability at least $1-2e^{-\iota}$,

$$\sum_{t=1}^{\tau}\|\nu_{t_0+t}\|^2 \leq c\cdot\psi^2(\tau+\iota), \tag{65}$$

From corollary 2, with the same probability of $1-2e^{-\iota}$,

$$\sum_{t=1}^{\tau}\|\nabla F(x_{t_0+t-1})\|^2 \leq \frac{2}{\eta}\left[F(x_{t_0})-F(x_{t_0+\tau})\right]+c\cdot\psi^2(\tau+\iota). \tag{66}$$

Combining above results, we have, with probability at least $1-2e^{-\iota}$,

$$\|x_{t_0+\tau}-x_{t_0}\|^2 \leq 4\eta\tau[F(x_{t_0})-F(x_{t_0+\tau})]+4c\cdot\eta^2\tau\psi^2(\tau+\iota). \tag{67}$$

Re-arranging the terms above, we obtain

$$F(x_{t_0+\tau})-F(x_{t_0}) \leq -\frac{1}{4\eta\tau}\|x_{t_0+\tau}-x_{t_0}\|^2+c\cdot\eta\psi^2(\tau+\iota). \tag{68}$$

According to the criterion for successful escape, we have $\|x_{t_0+\tau} - x_{t_0}\| \geq \mathcal{R}$. Then

$$F(x_{t_0+\tau}) - F(x_{t_0}) \leq -\frac{1}{4\eta\tau}\|x_{t_0+\tau} - x_{t_0}\|^2 + c \cdot \eta\psi^2(\tau + \iota) \tag{69}$$

$$\leq -\frac{\mathcal{R}^2}{4\eta\Gamma} + c \cdot \eta\psi^2(\Gamma + \iota) \tag{70}$$

$$\leq -\frac{s}{4\iota^3}\sqrt{\frac{\alpha^3}{\rho}} + \frac{2c \cdot \psi^2\iota}{s\sqrt{\rho\alpha}} \tag{71}$$

$$\leq -\frac{s}{8\iota^3}\sqrt{\frac{\alpha^3}{\rho}} = \Phi, \tag{72}$$

where the second to last inequality is from the fact that $s\eta\sqrt{\rho\alpha} = \frac{\rho\alpha}{M^2 s\mu^2} < 1$, and the last inequality follows from $\alpha \geq 4\sqrt{C}s\mu^2\psi$. □

B.6 Proof of Lemma 5

Proof of Lemma 5. By Corollary 3, for all t, $\nu_t \sim \mathrm{nSG}(C\sqrt{\sigma^2 + r^2 d})$. Since $\mathbb{E}[\nu_t] = 0$, by Definition 7, with probability at least $1 - \frac{\omega}{2T}$:

$$\|\nu_t\| \leq \sqrt{2}C\psi\sqrt{\log\frac{4T}{\omega}} \leq \chi. \tag{73}$$

Applying a union bound over $t \in [T]$ gives the desired result: with probability at least $1 - \omega/2$, $\|\hat{g}_t - \nabla F(x_{t-1})\| \leq \chi$ for all t. □

B.7 Proof of Lemma 6

Proof of Lemma 6. By Lemma 5, with probability at least $1 - \omega/2$, the gradient estimation error satisfies $\|\hat{g}_t - \nabla F(x_{t-1})\| \leq \chi$ for all $t \in [T]$. We analyze two cases based on whether the algorithm is in the escape phase.

Case 1: In escape phase. When $\|\hat{g}_t\| \leq 3\chi$, the escape process is triggered, implying $\|\nabla F(x_{t-1})\| \leq \alpha = 4\chi$. The average function decrease per step during a successful escape is at least:

$$\frac{\Phi}{\Gamma} = \frac{s^2\alpha^2\eta}{8\iota^4} = \frac{2\chi^2\eta}{s^2\mu^4}. \tag{74}$$

Case 2: Outside escape phase. When $\|\hat{g}_t\| > 3\chi$, we have $\|\nabla F(x_{t-1})\| \geq 2\chi$. Each PSGD step yields at least:

$$\frac{\eta}{2}(2\chi)^2 = 2\chi^2\eta > \frac{2\chi^2\eta}{s^2\mu^4}. \tag{75}$$

Thus, in either case, the function value decreases by at least $2\chi^2\eta/(s^2\mu^4)$ per step. Denoting $U := F_0 - F^*$, the number of effective descent steps is bounded by:

$$T_{\text{effective}} := \frac{Us^2\mu^4}{2\chi^2\eta}. \tag{76}$$

Next, consider the number of α-strict saddle points encountered. Each successful escape yields function decrease of at least Φ, so the total number of such escape phases is at most:

$$N_{\text{saddle}} := \frac{U}{\Phi} = \frac{8\iota^3 U}{s}\sqrt{\frac{\rho}{\chi^3}}. \tag{77}$$

By Corollary 1, each Γ-`descent` succeeds with probability at least 1/8, and we boost this to $1-\omega/2$ via the Q independent repetitions in every escape procedure. By Lemma 2 with failure probability $\omega_0 = \frac{\omega}{2N_{\text{saddle}}}$, we require:

$$Q = \frac{26}{5}\log\left(\frac{16\iota^3 U}{s\omega}\sqrt{\frac{\rho}{\chi^3}}\right). \tag{78}$$

Hence, the total number of PSGD steps (including all Γ-`descent` repetitions) is at most:

$$T \le T_{\text{effective}} \cdot Q = \frac{13Us^2\mu^4}{5\chi^2\eta}\log\left(\frac{16\iota^3 U}{s\omega}\sqrt{\frac{\rho}{\chi^3}}\right) = \tilde{O}\left(\frac{U}{\eta\chi^2}\right). \tag{79}$$

□

C Omitted Proofs in Section 6

C.1 Proof of Lemma 7

Proof of Lemma 7. Let $\tau(t)$ denote the most recent iteration (up to t) at which oracle $\mathcal{O}_1$ was used.

Case 1: If $t=\tau(t)$, then

$$\hat{g}_t = \mathcal{O}_1(x_{t-1},\mathcal{B}_t) + \xi_t. \tag{80}$$

Let $\zeta_t := \mathcal{O}_1(x_{t-1},\mathcal{B}_t) - \nabla F(x_{t-1})$, which is a zero-mean estimator with norm-subGaussian noise due to the G-Lipschitz condition:

$$\zeta_t \sim \text{nSG}\left(\frac{G\sqrt{\log d}}{\sqrt{b_1}}\right). \tag{81}$$

The noise term ξ_t is drawn from a Gaussian distribution:

$$\xi_t \sim \mathcal{N}\left(0, c_1\frac{G^2\log(1/\delta)}{b_1^2\epsilon^2}\mathbf{I}_d\right). \tag{82}$$

Thus, in this case, the oracle satisfies condition (2) with the desired bounds.

Case 2: If $t>\tau(t)$, then

$$\hat{g}_t = \mathcal{O}_1(x_{\tau(t)-1},\mathcal{B}_{\tau(t)}) + \xi_{\tau(t)} + \sum_{i=\tau(t)+1}^{t}\left(\mathcal{O}_2(x_{i-1},x_{i-2},\mathcal{B}_i)+\xi_i\right). \tag{83}$$

Let $\zeta_{\tau(t)} := \mathcal{O}_1(x_{\tau(t)-1},\mathcal{B}_{\tau(t)}) - \nabla F(x_{\tau(t)-1})$ and define

$$\zeta_i' := \mathcal{O}_2(x_{i-1},x_{i-2},\mathcal{B}_i) - \left(\nabla F(x_{i-1}) - \nabla F(x_{i-2})\right). \tag{84}$$

Then

$$\hat{g}_t - \nabla F(x_{t-1}) = \zeta_{\tau(t)} + \sum_{i=\tau(t)+1}^{t}\zeta_i' + \xi_{\tau(t)} + \sum_{i=\tau(t)+1}^{t}\xi_i. \tag{85}$$

By the M-smoothness assumption, we have

$$\zeta_i' \sim \text{nSG}\left(\frac{M\|x_{i-1}-x_{i-2}\|\sqrt{\log d}}{\sqrt{b_2}}\right), \tag{86}$$

and the privacy noise is drawn from

$$\xi_i \sim \mathcal{N}\left(0, c_2 \frac{M^2 \log(1/\delta)}{b_2^2 \epsilon^2} \|x_{i-1} - x_{i-2}\|^2 \mathbf{I}_d\right). \tag{87}$$

Since the algorithm ensures $\mathsf{drift}_t := \sum_{i=\tau(t)+1}^{t} \|x_{i-1} - x_{i-2}\|^2 \leq \kappa$, we can bound the noise as follows:

– From Corollary 3, the total norm-subGaussian parameter becomes:

$$\sigma \leq O\left(\sqrt{\left[\left(\frac{G\sqrt{\log d}}{\sqrt{b_1}}\right)^2 + \sum_{i=\tau(t)+1}^{t} \left(\frac{M\|x_{i-1} - x_{i-2}\|\sqrt{\log d}}{\sqrt{b_2}}\right)^2\right] \cdot \log d}\right) \tag{88}$$

$$\leq O\left(\sqrt{\frac{G^2 \log^2 d}{b_1} + \frac{M^2 \log^2 d}{b_2}\kappa}\right). \tag{89}$$

– By the property of Gaussian, the total privacy noise magnitude satisfies:

$$r \leq O\left(\sqrt{\frac{G^2 \log \frac{1}{\delta}}{b_1^2 \epsilon^2} + \sum_{i=\tau(t)+1}^{t} \left(\frac{M^2 \log \frac{1}{\delta}}{b_2^2 \epsilon^2} \|x_{t-1} - x_{t-2}\|^2\right)}\right) \tag{90}$$

$$\leq O\left(\sqrt{\frac{G^2 \log \frac{1}{\delta}}{b_1^2 \epsilon^2} + \frac{M^2 \log \frac{1}{\delta}}{b_2^2 \epsilon^2}\kappa}\right). \tag{91}$$

□

C.2 Proof of Lemma 8

Proof of Lemma 8. By the M-smoothness assumption and using the fact $\eta \leq \frac{1}{M}$, we apply the standard descent lemma:

$$\begin{aligned} F(x_t) - F(x_{t-1}) &\leq \langle \nabla F(x_{t-1}), x_t - x_{t-1} \rangle + \frac{M}{2}\|x_t - x_{t-1}\|^2 \\ &\leq \langle \nabla F(x_{t-1}) - \hat{g}_t, -\eta \cdot \hat{g}_t \rangle - \eta\|\hat{g}_t\|^2 + \frac{\eta}{2}\|\hat{g}_t\|^2 \\ &\leq \eta\|\nabla F(x_{t-1}) - \hat{g}_t\|\|\hat{g}_t\|_2 - \frac{\eta}{2}\|\hat{g}_t\|^2. \end{aligned}$$

By Lemma 5, with probability at least $1 - \omega/2$, we have $\|\nabla F(x_{t-1}) - \hat{g}_t\| \leq \chi$ for all t.

Now consider two cases:

Case 1: If $\|\nabla F(x_{t-1})\| \geq 4\chi$, then

$$\|\hat{g}_t\| \geq \|\nabla F(x_{t-1})\| - \chi \geq 3\chi \geq 3\|\nabla F(x_{t-1}) - \hat{g}_t\|, \tag{92}$$

yielding

$$F(x_t) - F(x_{t-1}) \leq -\frac{\eta}{6}\|\hat{g}_t\|^2. \tag{93}$$

Case 2: If $\|\nabla F(x_{t-1})\| \leq 4\chi$, then $\|\hat{g}_t\| \leq 5\chi$, and thus

$$F(x_t) - F(x_{t-1}) \leq 5\eta\chi^2. \tag{94}$$

Let $\mathcal{T} = \{t_1, t_2, \ldots, t_{|\mathcal{T}|}\}$ denote the set of iterations where model drift exceeds κ. For each pair of successive drift resets:

$$F(x_{t_{i+1}}) - F(x_{t_i}) \leq -\frac{1}{6\eta} \sum_{t=t_i+1}^{t_{i+1}} \eta^2 \|\hat{g}_t\|_2^2 + (t_{i+1} - t_i) 5\eta\chi^2 \tag{95}$$

$$\leq -\frac{1}{6\eta} \operatorname{drift}_{t_{i+1}} + (t_{i+1} - t_i) 5\eta\chi^2 \leq -\frac{1}{6\eta}\kappa + (t_{i+1} - t_i) 5\eta\chi^2. \tag{96}$$

Summing over i, we obtain:

$$F(x_{t_{|\mathcal{T}|}}) - F(x_{t_1}) \leq -\frac{|\mathcal{T}|}{6\eta}\kappa + 5T\eta\chi^2.$$

Since $F(\cdot)$ is upper bounded by U, we must have:

$$-U \leq -\frac{|\mathcal{T}|\kappa}{6\eta} + 5T\eta\chi^2, \tag{97}$$

which yields:

$$|\mathcal{T}| \leq O\left(\frac{U\eta}{\kappa} + \frac{T\eta^2\chi^2}{\kappa}\right) = O\left(\frac{U\eta}{\kappa}\right),$$

using $T = O(U/(\eta\chi^2))$. □

C.3 Proof of Theorem 2

Proof of Theorem 2. We first verify that the batch size settings b_1 and b_2 are feasible, i.e., the total number of data samples used remains $O(n)$. Recall from Lemma 8 that the number of rounds where drift exceeds the threshold is bounded by $|\mathcal{T}| = O(U\eta/\kappa)$, and the total number of steps is $T = O(U/(\eta\chi^2))$. Then:

$$b_1 \cdot |\mathcal{T}| + b_2 \cdot (T - |\mathcal{T}|) \leq b_1 \cdot |\mathcal{T}| + b_2 \cdot T \leq O(n), \tag{98}$$

under our settings of $b_1 = \frac{n\kappa}{2U\eta}$ and $b_2 = \frac{n\eta\chi^2}{2U}$. This confirms feasibility.

Since each sample is used only once, the overall (ϵ, δ)-differential privacy guarantee follows directly from the Gaussian mechanism and the parallel composition theorem.

We now derive the convergence error α via Theorem 1, which gives:

$$\alpha = O(\chi) = \tilde{O}(\psi) = \tilde{O}(\sqrt{\sigma^2 + r^2 d}), \tag{99}$$

where from Lemma 7:

$$\sigma^2 \leq \tilde{O}\left(\frac{G^2}{b_1} + \frac{M^2\kappa}{b_2}\right), \quad r^2 \leq \tilde{O}\left(\frac{G^2}{b_1^2\epsilon^2} + \frac{M^2\kappa}{b_2^2\epsilon^2}\right). \tag{100}$$

Substituting our settings $b_1 = \frac{n\kappa}{2U\eta}$ and $b_2 = \frac{n\eta\chi^2}{2U}$ into the expression, we get:

$$\alpha = \tilde{O}\left(\sqrt{\frac{G^2U\eta}{n\kappa} + \frac{G^2dU^2\eta^2}{n^2\epsilon^2\kappa^2} + \frac{M^2U\kappa}{n\eta\chi^2} + \frac{M^2dU^2\kappa}{n^2\epsilon^2\eta^2\chi^4}}\right) \tag{101}$$

$$= \tilde{O}\left(\sqrt{\frac{G^2U\sqrt{\rho\alpha}}{M^2n\kappa} + \frac{G^2dU^2\rho\alpha}{M^4n^2\epsilon^2\kappa^2} + \frac{M^4U\kappa}{\sqrt{\rho}n\alpha^{5/2}} + \frac{M^6dU^2\kappa}{\rho n^2\epsilon^2\alpha^5}}\right). \tag{102}$$

To isolate α, we take the largest among the resulting bounds:

$$\alpha = \tilde{O}\left(\max\left\{\left(\frac{G^2U\sqrt{\rho}}{M^2n\kappa}\right)^{2/3}, \frac{G^2dU^2\rho}{M^4n^2\epsilon^2\kappa^2}, \left(\frac{M^4U\kappa}{n\sqrt{\rho}}\right)^{2/9}, \left(\frac{M^6dU^2\kappa}{\rho n^2\epsilon^2}\right)^{1/7}\right\}\right). \tag{103}$$

Now set:

$$\kappa = \max\left\{\frac{G^{3/2}U^{1/2}\rho^{1/2}}{M^{5/2}n^{1/2}}, \frac{G^{14/15}d^{2/5}U^{4/5}\rho^{8/15}}{M^{34/15}(n\epsilon)^{4/5}}\right\}. \tag{104}$$

Substituting this into the above expression of α yields:

$$\alpha = \tilde{O}\left(\left(\frac{GUM}{n}\right)^{1/3} + \frac{G^{2/15}U^{2/5}M^{8/15}}{\rho^{1/15}}\left(\frac{\sqrt{d}}{n\epsilon}\right)^{2/5}\right) = \tilde{O}\left(\frac{1}{n^{1/3}} + \left(\frac{\sqrt{d}}{n\epsilon}\right)^{2/5}\right). \tag{105}$$

□

D Omitted Proofs in Section 7

D.1 Proof of Lemma 9

Proof of Lemma 9. Let $\tau(t)$ denote the most recent iteration at which oracle $\mathcal{O}_1$ was queried before or at iteration t.

Case 1: If $t = \tau(t)$, then the global estimator is

$$\hat{g}_t = \frac{1}{m}\sum_{j=1}^{m}\left(\mathcal{O}_1(x_{t-1}, \mathcal{B}_{j,t}) + \xi_{j,t}\right). \tag{106}$$

Each $\mathcal{O}_1(x_{t-1}, \mathcal{B}_{j,t})$ is an unbiased estimate of $\nabla F_j(x_{t-1})$. Let $\zeta_{j,t} := \mathcal{O}_1(x_{t-1}, \mathcal{B}_{j,t}) - \nabla F_j(x_{t-1})$, and define $\zeta_t := \frac{1}{m}\sum_j \zeta_{j,t}$ and $\xi_t := \frac{1}{m}\sum_j \xi_{j,t}$. Then,

$$\hat{g}_t - \nabla F(x_{t-1}) = \zeta_t + \xi_t. \tag{107}$$

Since f is G-Lipschitz, we have $\zeta_t \sim \mathrm{nSG}\left(\frac{G\sqrt{\log d}}{\sqrt{mb_1}}\right)$. Each $\xi_{j,t} \sim \mathcal{N}\left(0, c_1\frac{G^2\log(1/\delta)}{b_1^2\epsilon^2}\mathbf{I}_d\right)$, so their average satisfies:

$$\xi_t \sim \mathcal{N}\left(0, c_1\frac{G^2\log(1/\delta)}{mb_1^2\epsilon^2}\mathbf{I}_d\right). \tag{108}$$

Thus, in this case, the oracle satisfies condition (2) with the desired bounds.

Case 2: If $t > \tau(t)$, the global estimate is:

$$\hat{g}_t = \frac{1}{m}\sum_{j=1}^{m}\left(\mathcal{O}_1(x_{\tau(t)-1}, \mathcal{B}_{j,\tau(t)}) + \xi_{j,\tau(t)} + \sum_{i=\tau(t)+1}^{t}\left[\mathcal{O}_2(x_{i-1}, x_{i-2}, \mathcal{B}_{j,i}) + \xi_{j,i}\right]\right). \tag{109}$$

Let $\zeta_{j,\tau} := \mathcal{O}_1(x_{\tau(t)-1}, \mathcal{B}_{j,\tau(t)}) - \nabla F_j(x_{\tau(t)-1})$, and define:

$$\zeta'_{j,i} := \mathcal{O}_2(x_{i-1}, x_{i-2}, \mathcal{B}_{j,i}) - \left[\nabla F_j(x_{i-1}) - \nabla F_j(x_{i-2})\right]. \tag{110}$$

Then,

$$\hat{g}_t - \nabla F(x_{t-1}) = \zeta_{\tau(t)} + \sum_{i=\tau(t)+1}^{t}\zeta'_i + \xi_{\tau(t)} + \sum_{i=\tau(t)+1}^{t}\xi_i, \tag{111}$$

where $\zeta_{\tau(t)} := \frac{1}{m}\sum_j \zeta_{j,\tau(t)}$, $\zeta'_i := \frac{1}{m}\sum_j \zeta'_{j,i}$, and similarly for $\xi_{\tau(t)}$ and ξ_i. By the M-smoothness of f, we have:

$$\zeta'_i \sim \mathrm{nSG}\left(\frac{M\|x_{i-1}-x_{i-2}\|\sqrt{\log d}}{\sqrt{mb_2}}\right), \quad \xi_i \sim \mathcal{N}\left(0, c_2\frac{M^2\log(1/\delta)}{mb_2^2\epsilon^2}\|x_{i-1}-x_{i-2}\|^2\mathbf{I}_d\right). \tag{112}$$

Since the algorithm ensures that $\mathsf{drift}_t := \sum_{i=\tau(t)+1}^{t}\|x_{i-1}-x_{i-2}\|^2 \leq \kappa$, we obtain:

$$\sigma = \tilde{O}\left(\sqrt{\frac{G^2\log^2 d}{mb_1} + \frac{M^2\log^2 d}{mb_2}\kappa}\right), \quad r = \tilde{O}\left(\sqrt{\frac{G^2\log(1/\delta)}{mb_1^2\epsilon^2} + \frac{M^2\log(1/\delta)}{mb_2^2\epsilon^2}\kappa}\right). \tag{113}$$

□

D.2 Proof of Theorem 3

Proof of Theorem 3. We first verify that the total sample usage per client is $O(n)$. From Lemma 8, we have $|\mathcal{T}| = O(U\eta/\kappa)$ and $T = O(U/(\eta\chi^2))$. Using the settings:

$$b_1 = \frac{n\kappa}{2U\eta}, \quad b_2 = \frac{n\eta\chi^2}{2U}, \tag{114}$$

the total number of samples used per client is:

$$b_1 \cdot |\mathcal{T}| + b_2 \cdot (T - |\mathcal{T}|) \leq b_1 \cdot |\mathcal{T}| + b_2 \cdot T = O(n). \tag{115}$$

Differential privacy guarantees follows from the Gaussian mechanism and parallel composition, since each data point is used at most once.

Now for the error analysis. By Theorem 1:

$$\alpha = O(\chi) = \tilde{O}(\psi) = \tilde{O}(\sqrt{\sigma^2 + r^2 d}). \tag{116}$$

From Lemma 9:

$$\alpha = \tilde{O}\left(\sqrt{\frac{G^2}{mb_1} + \frac{G^2 d}{mb_1^2\epsilon^2} + \left(\frac{M^2}{mb_2} + \frac{M^2 d}{mb_2^2\epsilon^2}\right)\cdot\kappa}\right). \tag{117}$$

Substitute the expressions for b_1, b_2 into the bound and simplify, we get:

$$\alpha = \tilde{O}\left(\sqrt{\frac{G^2U\eta}{mn\kappa} + \frac{G^2dU^2\eta^2}{mn^2\epsilon^2\kappa^2} + \frac{M^2U\kappa}{mn\eta\chi^2} + \frac{M^2dU^2\kappa}{mn^2\epsilon^2\eta^2\chi^4}}\right) \tag{118}$$

$$= \tilde{O}\left(\sqrt{\frac{G^2U\sqrt{\rho\alpha}}{mM^2n\kappa} + \frac{G^2dU^2\rho\alpha}{mn^2\epsilon^2M^4\kappa^2} + \frac{M^4U\kappa}{mn\rho^{\frac{1}{2}}\alpha^{\frac{5}{2}}} + \frac{M^6dU^2\kappa}{mn^2\epsilon^2\rho\alpha^5}}\right). \tag{119}$$

To isolate α, we take the largest among the resulting bounds:

$$\alpha = \tilde{O}\left(\max\left\{\left(\frac{G^2U\sqrt{\rho}}{mM^2n\kappa}\right)^{2/3}, \frac{G^2dU^2\rho}{mn^2\epsilon^2M^4\kappa^2}, \left(\frac{M^4U\kappa}{mn\sqrt{\rho}}\right)^{2/9}, \left(\frac{M^6dU^2\kappa}{mn^2\epsilon^2\rho}\right)^{1/7}\right\}\right).$$

Now set:

$$\kappa = \max\left\{\frac{G^{3/2}\sqrt{\rho U}}{M^{5/2}\sqrt{mn}}, \frac{G^{14/15}d^{2/5}U^{4/5}\rho^{8/15}}{M^{34/15}(\sqrt{m}n\epsilon)^{4/5}}\right\} \tag{120}$$

Substituting this into the above expression of α yields:

$$\alpha = \tilde{O}\left(\left(\frac{GUM}{mn}\right)^{1/3} + \frac{G^{2/15}U^{2/5}M^{8/15}}{\rho^{1/15}}\left(\frac{\sqrt{d}}{\sqrt{m}n\epsilon}\right)^{2/5}\right) = \tilde{O}\left(\frac{1}{(mn)^{1/3}} + \left(\frac{\sqrt{d}}{\sqrt{m}n\epsilon}\right)^{2/5}\right). \tag{121}$$

□

D.3 Proof of Theorem 4

Proof of Theorem 4. The (ϵ, δ)-ICRL-DP guarantee follows directly from the Gaussian mechanism and the adaptive composition theorem, since each client adds independent Gaussian noise to both their gradient and Hessian estimates. Each local data point is used at most T times—once for each model iterate—and all messages sent to the server are privatized accordingly.

We now derive the error rate α guarantee for the output x_o. Let $\mathcal{S} := \bigsqcup_{j=1}^{m} S_j$ denote the full held-out evaluation dataset, and let x_p be an α-SOSP in the input to Algorithm 4. Define the aggregate gradient noise and Hessian noise as

$$\theta_p := \frac{1}{m}\sum_{j=1}^{m}\theta_{j,p}, \quad \mathbf{H}_p := \frac{1}{m}\sum_{j=1}^{m}\mathbf{H}_{j,p}. \tag{122}$$

Let $\sigma_1^2 = c_1\frac{G^2T\log(1/\delta)}{n^2\epsilon^2}$ and $\sigma_2^2 = c_2\frac{M^2dT\log(1/\delta)}{n^2\epsilon^2}$ denote the variances of the noise added to the gradient and Hessian components, respectively.

Gradient Estimation Error. For any $\mathcal{S}_j$ and x, $\nabla\hat{f}_{\mathcal{S}_j}(x) - \nabla F_j(x)$ is zero-mean and follows nSG $\left(\frac{2G}{\sqrt{n}}\right)$. By the G-Lipschitz assumption and norm-sub-Gaussian concentration (Lemma 11), we have with probability at least $1 - \omega'/8$:

$$\|\nabla F(x_p) - \nabla\hat{f}_{\mathcal{S}}(x_p)\| \le O\left(\frac{G\sqrt{\log(d/\omega')}}{\sqrt{mn}}\right). \tag{123}$$

Also, since $\theta_p \sim \mathcal{N}(0, \sigma_1^2/m)$, standard Gaussian concentration (Lemma 10) gives, with probability at least $1 - \omega'/8$:

$$\|\theta_p\| \le O\left(\frac{G\sqrt{dT\log(1/\delta)\log(1/\omega')}}{\sqrt{m}n\epsilon}\right). \tag{124}$$

Hessian Estimation Error. For any $j \in [m]$ and $z \in \mathcal{S}_j$, $\mathbb{E}[\nabla^2 f(x_p; z) - \nabla^2 F_j(x_p)] = 0$, and $\|\nabla^2 f(x_p; z) - \nabla^2 F_j(x_p)\|_2 \le 2M$ (due to M-smoothness). That is, each empirical Hessian term is $2M$-bounded in operator norm. Applying the matrix Bernstein inequality (Lemma 14), and using the assumption $mn \ge \frac{4}{9}\log(8d/\omega')$, we obtain with probability at least $1 - \omega'/8$:

$$\left\|\nabla^2\hat{f}_{\mathcal{S}}(x_p) - \nabla^2 F(x_p)\right\| \le O\left(M\sqrt{\frac{\log(d/\omega')}{mn}}\right). \tag{125}$$

For the added noise, since $\mathbf{H}_p$ consists of symmetric Gaussian matrices with variance σ_2^2/m, Lemma 15 gives, with probability at least $1-\omega'/8$:

$$\|\mathbf{H}_p\| \le O\left(\frac{Md\sqrt{T\log(1/\delta)\log(1/\omega')}}{\sqrt{m}n\epsilon}\right). \tag{126}$$

Verification for x_p. Combining the above estimates and using a union bound, with probability at least $1-\omega'/2$, we have:

$$\begin{aligned}
\|\nabla\bar{F}(x_p)\|_2 &\le \|\nabla F(x_p)\|_2 + \|\nabla\bar{F}(x_p)-\nabla F(x_p)\|_2 && (127)\\
&\le \|\nabla F(x_p)\|_2 + \|\nabla\hat{f}_{\mathcal{S}}(x_p)-\nabla F(x_p)\|_2 + \|\theta_p\|_2 && (128)\\
&\le \alpha + (\text{estimation error}) && (129)\\
&\le O\left(\alpha + \frac{G\log(d/\omega')}{\sqrt{mn}} + \frac{G\sqrt{dT\log(1/\delta)\log(1/\omega')}}{\sqrt{m}n\epsilon}\right), && (130)
\end{aligned}$$

and

$$\begin{aligned}
\lambda_{\min}\left(\nabla^2\bar{F}(x_p)\right) &\ge \lambda_{\min}\left(\nabla^2 F(x_p)\right) + \lambda_{\min}\left(\nabla^2\bar{F}(x_p)-\nabla^2 F(x_p)\right) && (131)\\
&\ge \lambda_{\min}\left(\nabla^2 F(x_p)\right) + \lambda_{\min}\left(\nabla^2\hat{f}_{\mathcal{S}}(x_p)-\nabla^2 F(x_p)\right) + \lambda_{\min}\left(\mathbf{H}_p\right) && (132)\\
&\ge -\sqrt{\rho\alpha} - \left\|\nabla^2 f(x_p;\mathcal{S})-\nabla^2 F(x_p)\right\|_2 - \|\mathbf{H}_p\|_2 && (133)\\
&\ge -\left(\sqrt{\rho\alpha} + (\text{estimation error})\right) && (134)\\
&\ge -O\left(\sqrt{\rho\alpha} + M\sqrt{\frac{\log(d/\omega')}{mn}} + \frac{Md\sqrt{T\log(1/\delta)\log(1/\omega')}}{\sqrt{m}n\epsilon}\right). && (135)
\end{aligned}$$

Hence, x_p will be selected with probability at least $1-\omega'/2$.

Guarantee for Output x_o. Let x_o be the output of Algorithm 4. By construction, it must satisfy:

$$\begin{aligned}
\|\nabla F(x_o)\|_2 &\le \|\nabla\bar{F}(x_o)\|_2 + \|\nabla F(x_o)-\nabla\bar{F}(x_o)\|_2 && (136)\\
&\le \|\nabla\bar{F}(x_o)\|_2 + \|\nabla F(x_o)-\nabla\hat{f}_{\mathcal{S}}(x_o)\|_2 + \|\xi_o\|_2, && (137)
\end{aligned}$$

and

$$\begin{aligned}
\lambda_{\min}(\nabla^2 F(x_o)) &\ge \lambda_{\min}(\nabla^2\bar{F}(x_o)) + \lambda_{\min}(\nabla^2 F(x_o)-\nabla^2\bar{F}(x_o)) && (138)\\
&\ge \lambda_{\min}(\nabla^2\bar{F}(x_o)) - \|\nabla^2 F(x_o)-\nabla^2\bar{F}(x_o)\|_2 && (139)\\
&\ge \lambda_{\min}(\nabla^2\bar{F}(x_o)) - \|\nabla^2 F(x_o)-\nabla^2\hat{f}_{\mathcal{S}}(x_o)\|_2 - \|H_o\|_2. && (140)
\end{aligned}$$

Using the same reasoning as above, applying the union bound again and using the fact that x_o is the output, we get that with probability at least $1-\omega'$, the following hold:

$$\|\nabla F(x_o)\| \le O\left(\alpha + \frac{G\log(d/\omega')}{\sqrt{mn}} + \frac{G\sqrt{dT\log(1/\delta)\log(1/\omega')}}{\sqrt{m}n\epsilon}\right), \tag{141}$$

and

$$\lambda_{\min}(\nabla^2 F(x_o)) \ge -O\left(\sqrt{\rho\alpha} + M\sqrt{\frac{\log(d/\omega')}{mn}} + \frac{Md\sqrt{T\log(1/\delta)\log(1/\omega')}}{\sqrt{m}n\epsilon}\right). \tag{142}$$

Finally, recalling that $T = O(1/\alpha^{2.5})$, and grouping the dependency on α, d, m, n, and ϵ, we conclude that x_o is an α'-SOSP with

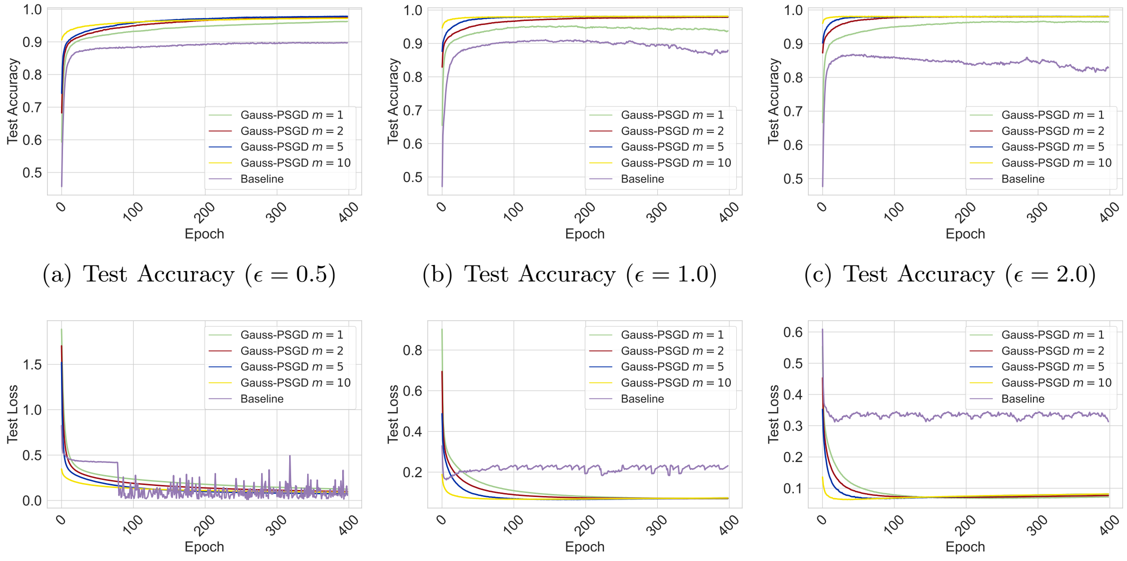
$$\alpha' = \tilde{O}\left(\alpha + \frac{1}{mn} + \frac{1}{\sqrt{mn}} + \frac{\alpha}{\sqrt{mn}} + \frac{\sqrt{d}}{\sqrt{m}n\epsilon\alpha^{5/4}} + \frac{d}{\sqrt{m}n\epsilon\alpha^{3/4}} + \frac{d^2}{mn^2\epsilon^2\alpha^{5/2}}\right), \qquad (143)$$

as claimed. □

E Experiments

Running Environments All experiments were conducted with the following computing infrastructure:

- OS: Ubuntu 22.04.4 LTS
- CPU: AMD EPYC 7513 32-Core Processor
- CPU Memory: 503GB
- GPU: NVIDIA RTX A6000 GPU
- GPU Memory: 48GB
- Programming language: Python 3.11.8
- Deep learning framework: Pytorch 2.2.2 + cuda 12.1



(a) Test Accuracy ($\epsilon = 0.5$)

(b) Test Accuracy ($\epsilon = 1.0$)

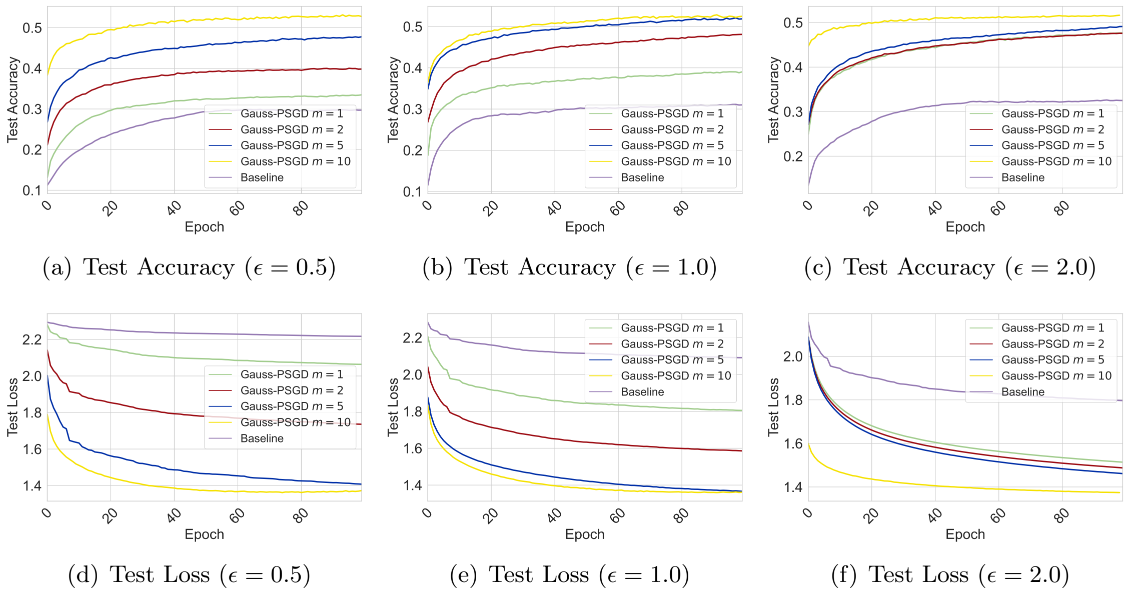
(c) Test Accuracy ($\epsilon = 2.0$)

(d) Test Loss ($\epsilon = 0.5$)

(e) Test Loss ($\epsilon = 1.0$)

(f) Test Loss ($\epsilon = 2.0$)

Figure 1: Comparison of learning performance for our Gauss-PSGD and the baseline method on **MNIST** dataset. **Top: Test accuracy** v.s. # epoch for varying privacy budget $\epsilon \in \{0.5, 1.0, 2.0\}$. **Bottom: Test loss** v.s. # epoch for varying privacy budget $\epsilon \in \{0.5, 1.0, 2.0\}$.



(a) Test Accuracy ($\epsilon = 0.5$)

(b) Test Accuracy ($\epsilon = 1.0$)

(c) Test Accuracy ($\epsilon = 2.0$)

(d) Test Loss ($\epsilon = 0.5$)

(e) Test Loss ($\epsilon = 1.0$)

(f) Test Loss ($\epsilon = 2.0$)

Figure 2: Comparison of learning performance for our Gauss-PSGD and the baseline method on **CIFAR-10** dataset. **Top: Test accuracy** v.s. # epoch for varying privacy budget $\epsilon \in \{0.5, 1.0, 2.0\}$. **Bottom: Test loss** v.s. # epoch for varying privacy budget $\epsilon \in \{0.5, 1.0, 2.0\}$.

Tasks and Datasets We conduct image classification tasks on two datasets: MNIST [26] and CIFAR-10 [24]. For each experiment, we set the number of training samples to $n = 6000$ and vary the number of clients m in $\{1, 2, 5, 10\}$, where $m = 1$ corresponds to the single-machine setting, while the others correspond to distributed learning scenarios. The test set consists of 10000 samples for both datasets.

Models We use a fully connected neural network with one hidden layer containing 128 units and ReLU activation. The loss function is the standard cross-entropy loss. The model is initialized using Kaiming initialization [18], with biases set to zero by default.

Algorithms We compare our proposed algorithm, which is abbreviated as **Gauss-PSGD**, against the baseline method from [29]. The hyperparameters for Gauss-PSGD are set as follows:

- Escape threshold $\chi = 0.01$
- Model drift threshold $\kappa = 0.1$
- Maximum escape steps $\Gamma = 10$
- Maximum repeat number of escape $Q = 3$

For all algorithms, we set the privacy parameters to $\delta = 10^{-5}$ and vary ϵ in $\{0.5, 1.0, 2.0\}$, corresponding to strong, medium, and weak privacy regimes, respectively. The learning rate is set to 0.001 for MNIST and 0.01 for CIFAR-10.

Evaluations We evaluate the performance of the implemented algorithms using two criteria: test accuracy and test loss. Both metrics are analyzed over training epochs to assess convergence and generalization performance.

Results The experimental results for the MNIST and CIFAR-10 datasets are shown in Fig. 1 and Fig.2, respectively. In each figure, we present test accuracy (top row) and test loss (bottom row) against the number of epochs for different privacy budgets ($\epsilon = 0.5, 1.0, 2.0$). From the experimental results, it can be seen that our proposed Gauss-PSGD consistently outperforms the baseline across all configurations. Specifically, Gauss-PSGD achieves higher test accuracy than the baseline for both datasets, with accuracy improving as ϵ increases due to weaker privacy constraints. The accuracy gap between Gauss-PSGD and the baseline widens in distributed settings ($m > 1$), highlighting the collaborative synergy of distributed learning and the robustness of Gauss-PSGD in handling data heterogeneity. Gauss-PSGD exhibits lower test loss compared to the baseline across all configurations. The rapid reduction in loss during the initial epochs indicates faster convergence, which holds true for both datasets and all privacy budgets. In conclusion, the results demonstrate that Gauss-PSGD achieves superior accuracy, faster convergence, and better scalability compared to the baseline.

F Broader Impact Statement

This paper advances the field of differentially private (DP) stochastic non-convex optimization by addressing key theoretical challenges in finding second-order stationary points (SOSP). Our contributions are particularly relevant for applications requiring strong privacy guarantees, including distributed learning with heterogeneous data. These advancements have practical implications for privacy-sensitive fields such as healthcare, finance, and large language models (LLMs), where data confidentiality is paramount.

By improving the efficiency and accuracy of DP optimization techniques, our work supports the development of machine learning systems that can operate on sensitive datasets without compromising privacy. This fosters greater trust in data-driven decision-making and encourages organizations to adopt privacy-preserving practices, enabling informed and responsible use of sensitive data.

Nevertheless, it is important to acknowledge the broader limitations inherent to DP-based learning algorithms, not just those specific to our work. Privacy-preserving methods often introduce trade-offs, such as reduced model accuracy compared to their non-private counterparts, which may impact decision-making in high-stakes applications.

Despite these challenges, we believe that advancing and responsibly applying privacy-preserving optimization techniques will have a positive societal impact. By enabling secure and ethical data analysis, our work contributes to the broader goal of building trustworthy AI/ML systems.