DPZero: Private Fine-Tuning of Language Models without Backpropagation

Liang Zhang[1], Bingcong Li[1], Kiran Koshy Thekumparampil[2], Sewoong Oh[3], and Niao He[1]

[1]Department of Computer Science, ETH Zurich
[2]Amazon Search
[3]Paul G. Allen School of Computer Science and Engineering, University of Washington
{liang.zhang, bingcong.li, niao.he}@inf.ethz.ch,
kkt@amazon.com, sewoong@cs.washington.edu

Abstract

The widespread practice of fine-tuning large language models (LLMs) on domain-specific data faces two major challenges in memory and privacy. First, as the size of LLMs continues to grow, the memory demands of gradient-based training methods via backpropagation become prohibitively high. Second, given the tendency of LLMs to memorize training data, it is important to protect potentially sensitive information in the fine-tuning data from being regurgitated. Zeroth-order methods, which rely solely on forward passes, substantially reduce memory consumption during training. However, directly combining them with standard differentially private gradient descent suffers more as model size grows. To bridge this gap, we introduce DPZero, a novel private zeroth-order algorithm with nearly dimension-independent rates. The memory efficiency of DPZero is demonstrated in privately fine-tuning RoBERTa and OPT on several downstream tasks. Our code is available at https://github.com/Liang137/DPZero.

1 Introduction

Fine-tuning pretrained large language models (LLMs), such as BERT [28, 80, 107], OPT [148], LLaMA [120, 121], and GPT [101, 14, 97, 96], achieves state-of-the-art performance in a wide array of downstream applications. However, two significant challenges persist in practical adoption: memory demands for gradient-based optimizers and the need to safeguard the privacy of domain-specific fine-tuning data.

As the memory requirement of fine-tuning LLMs is increasingly becoming a bottleneck, various approaches have been proposed, spanning from parameter-efficient fine-tuning (PEFT) [69, 58] to novel optimization algorithms [110, 3]. Since these methods rely on backpropagation to compute the gradients, which can be memory-intensive, a recent trend has emerged in developing algorithms that do not require backpropagation [11, 113, 55, 57, 99, 19]. Specifically for LLMs, Malladi et al. [87] introduced zeroth-order methods for fine-tuning, thereby eliminating the backward pass and freeing up the memory for gradients and activations. Utilizing a single A100 GPU (80 GiB memory), zeroth-order methods are capable of fine-tuning a 30-billion-parameter model, whereas first-order methods, even equipped with PEFT, fail to fit into the memory for a model with more than 6.7 billion parameters. This greatly expands the potential for deploying and fine-tuning LLMs even on personal devices.

On the other hand, empirical studies have highlighted the risk of LLMs inadvertently revealing sensitive information from their fine-tuning datasets [91, 143, 90, 85]. Such privacy concerns are pronounced especially when users opt to fine-tune LLMs on datasets of their own. Notably, the expectation that machine learning models should not compromise the confidentiality of their contributing entities is codified into legal frameworks [126]. Differential privacy (DP) [33] is a widely accepted mathematical framework for ensuring privacy by preventing attackers from identifying participating entities [112]. Consequently, the development of methods that fine-tune LLMs under differential privacy is of pressing necessity [71, 140, 54, 16, 29]; however, most efforts so far have focused on first-order algorithms.

Motivated by the memory-hungry nature and privacy concerns in fine-tuning LLMs, we investigate zeroth-order methods that guarantee differential privacy for solving the following stochastic optimization problem:

$$\min_{x\in\mathbb{R}^d} F_S(x) \;:=\; \frac{1}{n}\sum_{i=1}^{n} f(x;\xi_i)\;, \tag{1}$$

where $S=\{\xi_i\}_{i=1}^n$ is the training data, $x\in\mathbb{R}^d$ is the model weight, the loss $f(x;\xi_i)$ is Lipschitz for each sample ξ_i, and the averaged loss $F_S(x)$ is smooth and possibly nonconvex. In theory, previous work on both differentially private optimization [8] and zeroth-order optimization [32] indicated that their convergence guarantees depend explicitly on the dimension d. Such dimension dependence becomes problematic in the context of LLMs with d scaling to billions. In practice, and somewhat surprisingly, empirical studies on the fine-tuning of LLMs using zeroth-order methods [87] and DP first-order methods [140, 71, 70] have shown that the performance degradation due to the large model size is marginal. For example, Yu et al. [140] showed that the performance drop due to privacy is smaller for larger architectures. A 345 million-sized GPT-2-Medium, fine-tuned with $(\varepsilon=6.8,\delta=10^{-5})$-DP, showcases a modest drop of 5.1 in BLEU score [98] (compared to a non-private model of the same size and architecture), whereas a larger GPT-2-XL with 1.5 billion parameters exhibits smaller cost in test performance, i.e., 4.3 BLEU score under the same privacy budget.

This gap between theory and practice has been linked to the presence of low-rank structures in the fine-tuning of pretrained LLMs [87, 70]. Empirical evidence suggests that fine-tuning occurs within a low-dimensional subspace [105, 51, 44, 68]: 200 dimensions for RoBERTa with 355 million parameters [2] and 100 dimensions for PEFT on DistilRoBERTa with 7 million parameters [70]. In such cases where the intrinsic dimension is small, zeroth-order methods are known to achieve dimension-independent convergence rate [87] and private first-order methods are also known to achieve dimension-independent guarantees [86, 70].

Given the significance of fine-tuning LLMs on domain-specific datasets, we ask the following fundamental question: *Can we achieve a dimension-independent rate both under differential privacy and with access only to the zeroth-order oracle?* Our contributions are summarized below.

- We first show that the straightforward approach — that combines DP first-order methods with zeroth-order gradient estimators (Algorithm 1) — exhibits an undesirable dimension dependence in the convergence guarantees, even when the effective rank of the problem does not scale with the dimension (Theorems 1 and 2 in Section 3). There are two root causes. First, the standard practice of choosing the clipping threshold to be the maximum norm of the estimated sample gradient leads to an unnecessarily large threshold. Next, this choice of the clipping threshold forces the addition of a large noise to ensure privacy, and Algorithm 1 adds that noise in all d directions.
- We present DPZero (Algorithm 2), the first nearly dimension-independent DP zeroth-order method for stochastic optimization. Its convergence guarantee depends on the effective rank of the problem (specified in Assumption 3.5) and exhibits logarithmic dependence on the dimension d (Theorem 3 in Section 4). This builds upon two insights. First, the direction of the estimated gradient is a public information and does not need to be private; it is sufficient to make only the magnitude of the estimated gradient private, which is a scalar value. Next, we introduce a tighter analysis that allows us to choose a significantly smaller clipping threshold, leveraging the fact that the typical norm of the estimated gradient is much smaller than its maximum.
- We verify the effectiveness of DPZero in both synthetic examples and private fine-tuning tasks on RoBERTa [80] and OPT [148]. In contrast to first-order algorithms that demand extensive effort for the efficient implementation of per-sample gradient clipping [71, 54, 16], DPZero offers the advantage of near-zero additional costs compared to non-private zeroth-order methods [87]. Our empirical results validate theoretical findings, revealing only a slight performance decrement for DPZero even with large model sizes.

1.1 Related Works

We build upon exciting advances in zeroth-order optimization and differentially private optimization, which we survey here. Notably, DPZero is inspired by new empirical and theoretical findings showing that fine-tuning LLMs does not suffer in high-dimensions when using zeroth-order methods in Malladi et al. [87] or using private first-order optimization in Li et al. [70]. A more comprehensive overview is deferred to Appendix A.

Zeroth-order optimization. Nesterov and Spokoiny [94] pioneered the formal analysis of the convergence rate of zeroth-order methods, i.e., zeroth-order (stochastic) gradient descent (ZO-SGD) that replaces gradients in SGD by their zeroth-order estimators. Their findings are later refined by several works [43, 108, 73]. These well-established results indicate a runtime complexity $\mathcal{O}(d)$ worse than first-order methods. Such dimension dependence of zeroth-order methods is proven inevitable without additional structures [134, 32].

There are several recent works that relax the dimension dependence in zeroth-order methods leveraging problem structures. Balasubramanian and Ghadimi [7] demonstrated that ZO-SGD can directly identify the sparsity of the problem and proved a dimension-independent rate when the support of gradients remains unchanged. Yue et al. [141] and Malladi et al. [87] relaxed the dependence on dimension d to a quantity related to the trace of the loss's Hessian.

Differentially private optimization. Previous works on DP optimization mostly center around first-order methods. When the problem is nonconvex, i.e., the setting of our interest, differentially private (stochastic) gradient descent (DP-GD) achieves a rate of $\mathcal{O}(\sqrt{d\log(1/\delta)}/(n\varepsilon))$ on the squared norm of the gradient [130, 150]. We show that DPZero matches this rate with access only to the zeroth-order oracle in Theorem 3. Given access to the first-order oracle, it has been recently shown that such rate can be improved to $\mathcal{O}((\sqrt{d\log(1/\delta)}/(n\varepsilon))^{4/3})$ leveraging momentum [122] or variance reduction techniques [4].

Early works established dimension-independent rates when the gradients lie in some fixed low-rank subspace [60, 116]. Closest to our result is Song et al. [116], which demonstrated that the rate of DP-GD for smooth nonconvex optimization can be improved to $\mathcal{O}(\sqrt{r\log(1/\delta)}/(n\varepsilon))$ for generalized linear models (GLMs) with a rank-r feature matrix. DPZero matches this result with access only to the zeroth-order oracle in Theorem 3 for more general problems beyond low-rank GLMs. Our result is inspired by Li et al. [70] that introduced a relaxed Lipschitz condition for the gradients and provided dimension-free bounds when the loss is convex and the relaxed Lipschitz parameters decay rapidly. Similarly, Ma et al. [86] suggested that the dependence on d in the utility upper bound for DP stochastic convex optimization can be improved.

Literature on DP optimization beyond first-order methods remains less explored. Recently, Zhang et al. [147] studied the problem of private zeroth-order nonsmooth nonconvex optimization and achieved a rate that depends on the dimension d. As far as we are aware, no prior studies have addressed the challenge of deriving a dimension-independent rate in DP zeroth-order optimization.

After the workshop version of our paper [146] was released, Tang et al. [117] concurrently discovered the same algorithm as DPZero (up to a minor difference in how u_t is drawn) and showed empirical benefits when applied to fine-tuning OPT models but without theoretical analysis. Also building upon the workshop version of our paper, Liu et al. [81] introduced DP-ZOSO, a stage-wise zeroth-order method with an additional quadratic regularizer. With extra hyper-parameters to be tuned, DP-ZOSO demonstrates further empirical gain over DPZero. However, Liu et al. [81] only provided *dimension-dependent* guarantees.

2 Preliminaries

Notation. We use $\|\cdot\|$ for the Euclidean norm and define $\|v\|_W^2 = v^\top W v$ for a square matrix W. $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d \,|\, \|x\| = 1\}$ denotes the unit sphere in $\mathbb{R}^d$, and $\eta\,\mathbb{S}^{d-1}$ is the sphere of radius $\eta > 0$. A function $p : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz if $|p(x_1) - p(x_2)| \le L\|x_1 - x_2\|, \forall x_1, x_2$. A function $q : \mathbb{R}^d \to \mathbb{R}$ is ℓ-smooth if it is differentiable and $\|\nabla q(x_1) - \nabla q(x_2)\| \le \ell\|x_1 - x_2\|$. The trace of a square matrix J is denoted by $\mathrm{Tr}(J)$. A symmetric real matrix $M \succeq 0$ if it is positive semi-definite. The clipping operation is defined to be $\mathrm{clip}_C(x) = x\,\min\{1, C/\|x\|\}$ given $C > 0$. The notation $\tilde{\mathcal{O}}(\cdot)$ hides additional logarithmic terms.

2.1 Differential Privacy

Definition 2.1 (Differential Privacy [33, 34])**.** Two datasets $S = \{\xi_i\}_{i=1}^n$ and $S' = \{\xi_i'\}_{i=1}^n$ are *neighboring* if $\max\{|S \setminus S'|, |S' \setminus S|\} = 1$, and we denote it by $S \sim S'$. For prescribed $\varepsilon > 0$ and $\delta \in (0, 1)$, an algorithm $\mathcal{A}$ is said to satisfy (ε, δ)-*differential privacy* (DP) if $\mathbb{P}(\mathcal{A}(S) \in \mathcal{B}) \le e^\varepsilon \mathbb{P}(\mathcal{A}(S') \in \mathcal{B}) + \delta$ for all $S \sim S'$ and all measurable set $\mathcal{B}$ in the range of $\mathcal{A}$.

To ensure DP while solving the optimization problem in Eq. (1), first-order approaches, such as DP-GD, update via $x_{t+1} \leftarrow x_t - \alpha((1/n)\sum_{i=1}^n \mathrm{clip}_C(\nabla f(x_t; \xi_i)) + z_t)$; see e.g., [115, 1]. Through the following

composition lemma [62, Theorem 4.3], the privacy for entire T updates is secured by the per-sample clipping operation that ensures finite sensitivity of $\Delta = 2C/n$ together with the Gaussian noise z_t.

Lemma 2.2 (Advanced Composition)**.** *Let $\mathcal{A}$ be some randomized algorithm operating on a dataset S and outputting a vector in $\mathbb{R}^d$. If $\mathcal{A}$ has sensitivity $\Delta := \sup_{S\sim S'} \|\mathcal{A}(S) - \mathcal{A}(S')\|$, the mechanism that adds Gaussian noise $\mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ with variance $\sigma^2 = (2\Delta\sqrt{2T\log(e + (\varepsilon/\delta))}/\varepsilon)^2$ satisfies (ε, δ)-DP under T-fold adaptive composition for any $\varepsilon > 0$ and $\delta \in (0, 1)$.*

2.2 Zeroth-Order Optimization

When the gradient is expensive to compute, zeroth-order methods are useful for optimizing Eq. (1). For example, the two-point gradient estimator below requires only two evaluation of function values [108]

$$g_\lambda(x; \xi_i) \;:=\; \frac{f(x + \lambda u; \xi_i) - f(x - \lambda u; \xi_i)}{2\lambda} u \,, \tag{2}$$

where u is sampled uniformly from the Euclidean sphere $\sqrt{d}\,\mathbb{S}^{d-1}$ and $\lambda > 0$ is the smoothing parameter [139, 31]. A common approach to generate u is to set $u = \sqrt{d}\, z/\|z\|$, with z sampled from the standard multivariate Gaussian $\mathcal{N}(0, \mathrm{I}_d)$ [92, 89]. We refer to $g_\lambda(x; \xi)$ as the *zeroth-order gradient* (estimator) in the sequel. The results in this paper can be directly extended to other zeroth-order gradient estimators, e.g., any u satisfying $\mathbb{E}[uu^\top] = \mathrm{I}_d$ [32], the one-point estimator [39], and the directional derivative [94].

3 DP-GD with Zeroth-Order Gradients Suffers in High Dimensions

In this section, we show that the direct integration of zeroth-order gradient estimators in Eq. (2) into DP-GD, which we term DPGD-0th, leads to undesirable dimension dependence in the error rate. Such dependence persists even under a low effective rank assumption.

3.1 Direct Integration Leads to an $\mathcal{O}(d^{3/2})$ Rate

We present in Algorithm 1 the straightforward private zeroth-order approach that substitutes the gradients in DP-GD with zeroth-order estimators $g_\lambda(x_t; \xi_i)$ in Eq. (2).

The privacy guarantee follows from standard DP-GD analysis, and the utility guarantee on the squared gradient norm is derived from classical techniques for analyzing zeroth-order methods [94]. Before presenting the convergence result, we make the following standard assumption, which is common in nonconvex DP optimization [130, 131, 122].

Assumption 3.1. The loss $f(x; \xi)$ is L-Lipschitz for every ξ. The average loss $F_S(x)$ is ℓ-smooth for every given dataset S, and its minimum $F_S^* := \min_{x\in\mathbb{R}^d} F_S(x)$ is finite.

Theorem 1. *For any $\varepsilon > 0$ and $\delta \in (0, 1)$, Algorithm 1 is (ε, δ)-DP. Under Assumption 3.1, its output x_τ satisfies that*

$$\mathbb{E}[\|\nabla F_S(x_\tau)\|^2] \;\le\; 16\Big((F_S(x_0) - F_S^*)\,\ell + 2L^2\Big)\frac{d\sqrt{d\log(e + (\varepsilon/\delta))}}{n\varepsilon}, \tag{3}$$

with the choice of parameters

$$\alpha = \frac{1}{4\ell d}, \quad T = \frac{n\varepsilon}{\sqrt{d\log(e + (\varepsilon/\delta))}}, \quad \lambda \le \frac{4L}{\ell d}\Big(\frac{\sqrt{d\log(e + (\varepsilon/\delta))}}{n\varepsilon}\Big)^{1/2}, \quad C = Ld.$$

The total number of zeroth-order gradient computations is $nT = \mathcal{O}(n^2/\sqrt{d})$.

Remark 3.2. Theorem 1 demonstrates that directly combining DP-GD with zeroth-order gradients leads to an $\mathcal{O}(d^{3/2})$ error complexity, which is $\mathcal{O}(d)$ worse than first-order DP approaches [130].

Algorithm 1 DP-GD with 0th-order gradients (DPGD-0th)

Input: Dataset $S = \{\xi_1, \cdots, \xi_n\}$, initialization $x_0 \in \mathbb{R}^d$, number of iterations T, stepsize $\alpha > 0$, smoothing parameter $\lambda > 0$, clipping threshold $C > 0$, privacy parameters $\varepsilon > 0, \delta \in (0, 1)$.

1: **for** $t = 0, 1, \cdots, T-1$ **do**

2: Sample u_t uniformly at random from the Euclidean sphere $\sqrt{d}\,\mathbb{S}^{d-1}$ and for all $i = 1, \cdots, n$ compute

$$g_\lambda(x_t; \xi_i) \leftarrow \frac{f(x_t + \lambda u_t; \xi_i) - f(x_t - \lambda u_t; \xi_i)}{2\lambda} u_t.$$

3: Sample $z_t \in \mathbb{R}^d$ randomly from the multivariate Gaussian distribution $\mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ with variance $\sigma = 4C\sqrt{2T\log(e + (\varepsilon/\delta))}/(n\varepsilon)$ and update

$$x_{t+1} \;\leftarrow\; x_t - \alpha\Big(\frac{1}{n}\sum_{i=1}^{n} \mathrm{clip}_C(g_\lambda(x_t; \xi_i)) + z_t\Big).$$

Output: x_τ for τ sampled uniformly at random from $\{0, 1, \cdots, T-1\}$.

Remark 3.3. Three sources contribute to the dependence in d: the squared norm of the zeroth-order gradient estimator $\mathbb{E}[\|(1/n)\sum_{i=1}^n g_\lambda(x, \xi_i)\|^2] = \mathcal{O}(d\,\|\nabla F_S(x)\|^2)$ when taking $\lambda \to 0$ for simplicity, the clipping threshold $C = \mathcal{O}(d)$, and the norm of the privacy noise $\mathbb{E}[\|z_t\|^2] = \mathcal{O}(d\,C^2) = \mathcal{O}(d^3)$. The standard analysis of one-step update gives

$$\mathbb{E}[F_S(x_{t+1})] \;\le\; \mathbb{E}[F_S(x_t)] - \frac{\alpha}{2}\big(1 - 2d\,\ell\alpha\big)\,\mathbb{E}[\|\nabla F_S(x_t)\|^2] + c\,\alpha^2\,d^3, \tag{4}$$

where c is a constant that depends on problem parameters other than α and d; see Eq. (12) for details. A small enough step size, $\alpha < 1/(2\ell d)$, is required to make the second term negative, where the dependence in d comes from $\mathbb{E}[\|(1/n)\sum_{i=1}^n g_\lambda(x, \xi_i)\|^2]$. The dependence on d^3 in the last term arises from $\mathbb{E}[\|z_t\|^2]$, which leads to the $\mathcal{O}(d^{3/2})$ rate in Eq. (3) after balancing error terms. Detailed proofs can be found in Appendix D.

Remark 3.4. The choice of the clipping threshold $C = Ld$ ensures that clipping does not happen with probability one, which is a common choice in the theoretical analysis of private optimization algorithms [8, 9, 130]. This follows from the fact that, for L-Lipschitz $f(x; \xi)$, the zeroth-order gradient is upper bounded by $\|g_\lambda(x; \xi)\| \le Ld$ almost surely. Selecting the clipping threshold without knowledge of this upper bound remains an active research topic [23, 138, 36, 66, 149].

3.2 Rate Improves to $\mathcal{O}(d)$ under Low Effective Rank

Here, under the low-dimensional structures in fine-tuning LLMs (cf. Section 1), we demonstrate improved performance for Algorithm 1. Unfortunately, a linear dependence in d still persists even under the low effective rank structure.

Assumption 3.5. The function $f(x; \xi)$ is L-Lipschitz and ℓ-smooth for every ξ. The average function $F_S(x)$ is twice differentiable with $-H \preceq \nabla^2 F_S(x) \preceq H$ for any $x \in \mathbb{R}^d$, and its minimum $F_S^* := \min_{x \in \mathbb{R}^d} F_S(x)$ is finite. Here, the real-valued $d \times d$ matrix $H \succeq 0$ satisfies that $\|H\|_2 \le \ell$ and $\mathrm{Tr}(H) \le r\|H\|_2$. We refer to r as the effective rank or the intrinsic dimension of the problem.

Assumption 3.5 boils down to Assumption 3.1 if $r = d$. This is because $-H' \preceq \nabla^2 F_S(x) \preceq H', \forall\, x \in \mathbb{R}^d$ and $H' = \ell\,\mathrm{I}_d$ imply that $\|H'\|_2 \le \ell$ and $\mathrm{Tr}(H') \le d\|H'\|_2$. With $r < d$, this assumption reflects the additional structures encoded in the Hessian matrix. While Assumption 3.5 naturally holds for low-rank Hessians, it covers more general cases. For example, the assumption is satisfied with $r = \mathcal{O}(\log d) \ll d$ in the case of a full-rank matrix H, with its i-th largest eigenvalue being ℓ/i for $1 \le i \le d$.

Similar assumptions have been made to relax the dimension dependence in zeroth-order optimization in the limit $\lambda \to 0$ [87] and also for DP first-order optimization when the objective is smooth and convex [86]. However, even under Assumption 3.5, DPGD-0th (Algorithm 1) still suffers from a linear dependence in d in its error rate, as presented below. A proof is provided in Appendix D.

Theorem 2. *For any $\varepsilon > 0$ and $\delta \in (0,1)$, Algorithm 1 is (ε, δ)-DP. Under Assumption 3.5, its output x_τ satisfies that*

$$\mathbb{E}[\|\nabla F_S(x_\tau)\|^2] \;\le\; 16\Big((F_S(x_0) - F_S^*)\,\ell + 2L^2\Big)\frac{d\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}, \tag{5}$$

with the choice of parameters

$$\alpha = \frac{1}{4\ell(r+2)}, \quad T = \frac{n(r+2)\varepsilon}{d\sqrt{r\log(e+(\varepsilon/\delta))}}, \quad \lambda \le \frac{4L}{\ell d}\Big(\frac{\sqrt{r\log(e+(\varepsilon/\delta))}}{n\varepsilon}\Big)^{1/2}, \quad C = Ld.$$

The total number of zeroth-order gradient computations is $nT = \mathcal{O}(n^2\sqrt{r}/d)$.

Remark 3.6. Comparing to Remark 3.3, both the zeroth-order gradient, $\mathbb{E}[\|(1/n)\sum_{i=1}^n g_\lambda(x_t;\xi_i)\|_H^2]$, and the DP noise, $\mathbb{E}[\|z_t\|_H^2]$, decrease by a factor of $\mathcal{O}(r/d)$ under low effective rank. This is made precise in Lemma C.1. As a result, the one-step update analysis can be tightened as

$$\mathbb{E}[F_S(x_{t+1})] \;\le\; \mathbb{E}[F_S(x_t)] - \frac{\alpha}{2}\big(1 - 2(r+2)\ell\alpha\big)\,\mathbb{E}[\|\nabla F_S(x_t)\|^2] + c\,\alpha^2\, r\, d^2. \tag{6}$$

Comparing to the RHS of Eq. (4), it achieves an improved dependence in d. However, the third term in Eq. (6) is still at $\mathcal{O}(d^2)$ due to the clipping threshold $C = \mathcal{O}(d)$. Consequently, even when the effective rank r is small, Eq. (5) still grows linearly in d.

4 DPZero: Nearly Dimension-Independent Private Zeroth-Order Optimization

A straightforward combination of DP-GD and zeroth-order methods has a large dimension dependence. Our novel DPZero overcomes this issue with two key insights elaborated below.

Scalar privacy noise. By decoupling zeroth-order gradients in Eq. (2) into direction and magnitude, our key observation is that the direction, u_t, is public knowledge, and we only need to make the magnitude private. Privacy can be guaranteed by clipping the finite-difference, $(f(x_t + \lambda u_t;\xi_i) - f(x_t - \lambda u_t;\xi_i))/(2\lambda)$, and then adding a *scalar* noise z_t; see line 3 of Algorithm 2. This change, when applied to Algorithm 1, can significantly improve the rate in Eq. (5) by a factor of $d^{1/2}$.

Tighter clipping threshold. Another factor of $d^{1/2}$ improvement originates from a tighter analysis on the upper bound of the finite-difference term. Although its worst-case upper bound scales with the dimension d, this only happens with an exponentially small probability over the randomness of u_t. As proved in Eq. (16) in Appendix E, the size of the finite-difference is

$$\frac{|f(x_t + \lambda u_t;\xi_i) - f(x_t - \lambda u_t;\xi_i)|}{2\lambda} \;\le\; |u_t^\top \nabla f(x_t;\xi_i)| + \frac{\ell}{2}\lambda d,$$

where we use the assumption that each $f(x;\xi)$ is ℓ-smooth. When u_t is sampled from the sphere $\sqrt{d}\,\mathbb{S}^{d-1}$, a tail bound (part (ii) of Lemma C.1 in the appendix) implies that

$$\mathbb{P}\left(\left|u_t^\top \nabla f(x_t;\xi_i)\right| \ge C\right) \le 2\sqrt{2\pi}\,\exp\Big(-\frac{C^2}{8L^2}\Big).$$

By selecting the smoothing parameter λ to be sufficiently small, a careful choice of $C = \tilde{\mathcal{O}}(L)$, which is nearly independent of d, can ensure that clipping does not occur with a high probability. This choice is significantly smaller than the worst-case clipping threshold of $Ld^{1/2}$. The main technical challenge is that we need to analyze the algorithm given the event that clipping does not happen. The choice of drawing u_t from the uniform distribution over the sphere, together with corresponding tail bounds in Appendix C, allows us to prove the following nearly dimension-independent bound under the low effective rank structure in Assumption 3.5. A proof is provided in Appendix E.

Algorithm 2 DPZero

Input: Dataset $S = \{\xi_1, \cdots, \xi_n\}$, initialization $x_0 \in \mathbb{R}^d$, number of iterations T, stepsize $\alpha > 0$, smoothing parameter $\lambda > 0$, clipping threshold $C > 0$, privacy parameters $\varepsilon > 0, \delta \in (0,1)$.

1: **for** $t = 0, 1, \cdots, T-1$ **do**

2: Sample u_t uniformly at random from the Euclidean sphere $\sqrt{d}\,\mathbb{S}^{d-1}$.

3: Sample a scalar $z_t \in \mathbb{R}$ randomly from the univariate Gaussian distribution $\mathcal{N}(0, \sigma^2)$ with variance $\sigma = 4C\sqrt{2T\log(e + (\varepsilon/\delta))}/(n\varepsilon)$ and update the parameter

$$x_{t+1} \leftarrow x_t - \alpha\Big(\frac{1}{n}\sum_{i=1}^{n} \mathrm{clip}_C\Big(\frac{f(x_t + \lambda u_t; \xi_i) - f(x_t - \lambda u_t; \xi_i)}{2\lambda}\Big) + z_t\Big)u_t.$$

Output: x_τ for τ sampled uniformly at random from $\{0, 1, \cdots, T-1\}$.

Theorem 3. *For any $\varepsilon > 0$ and $\delta \in (0,1)$, Algorithm 2 is (ε, δ)-DP. Under Assumption 3.5, suppose $\max_{0\le t\le T}|F_S(x_t)| \le B$, the output x_τ satisfies that*

$$\mathbb{E}[\|\nabla F_S(x_\tau)\|^2] \;\le\; \Big(64\Big((F_S(x_0) - F_S^*)\,\ell + \tilde{L}^2\Big) + 2L^2\Big)\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}, \tag{7}$$

where we define

$$\tilde{L}^2 = L^2\,\log\Big(\frac{2\sqrt{2\pi}\,n^3\varepsilon^2(r+2)(d + 8\ell B(r+2)/L^2)}{r\log(e + (\varepsilon/\delta))}\Big),$$

and choose the parameters to be

$$\alpha = \frac{1}{4\ell(r+2)}, \quad T = \frac{n(r+2)\varepsilon}{4\sqrt{r\log(e + (\varepsilon/\delta))}}, \quad C = 4\tilde{L},$$
$$\lambda \le \frac{1}{\ell d}\min\Big\{4(2-\sqrt{2})\tilde{L},\; \frac{L}{\sqrt{d}}\Big(\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}\Big)^{\frac{1}{2}}\Big\}.$$

The total number of zeroth-order gradient computations is $nT = \mathcal{O}(n^2\sqrt{r})$.

Remark 4.1. Algorithm 2 is nearly dimension-independent, given its logarithmic dependence on d. To the best of our knowledge, this is the first zeroth-order DP method that is nearly dimension-independent. This feature is significantly beneficial for fine-tuning pretrained LLMs where the effective rank has been observed to be quite small [2, 70]. When $r = d$, our rate in Eq. (7) nearly matches that of the best known achievable bound of the first-order method DP-GD for smooth nonconvex losses [130]. When the effective rank r is smaller, this algorithm achieves $\tilde{\mathcal{O}}(\sqrt{r\log(1/\delta)}/(n\varepsilon))$ squared gradient norm. Similar dimension-free error rate is established for DP-GD on unconstrained generalized linear losses [116], with a dependence on the rank of the feature matrix. Table 1 provides a summary on how DPZero depends on dimension d and effective rank r.

Table 1: The dependence of the error rate on dimension d and effective rank r shows that the proposed DPZero (Algorithm 2) significantly outperforms DPGD-0th (Algorithm 1) and achieves performance close to the popular first-order method, DP-GD, on both scenarios with and without a low-effective rank assumption. Note that the error rates of zeroth and first-order DP methods are achieved with different number of iterations.

	without Assumption 3.5	with Assumption 3.5
DPGD-0th	$\mathcal{O}(d\sqrt{d})$	$\mathcal{O}(d\sqrt{r})$
DPZero	$\mathcal{O}((\log d)\sqrt{d})$	$\mathcal{O}((\log d)\sqrt{r})$
DP-GD	$\mathcal{O}(\sqrt{d})$	$\mathcal{O}(\sqrt{r})$

Remark 4.2. The RHS of Eq. (7) improves upon Eq. (5) of Algorithm 1 by a factor of d. Simplifying our analysis in Eq. (22) and conditioned on the event that the clipping does not happen, we get a similar one-step update analysis as Eq. (6) (see Eq. (22) and (23) for a precise inequality). However, since the privacy noise z_t is a scalar and the clipping threshold has been reduced, we have that $\mathbb{E}[\|z_t u_t\|_H^2] = \tilde{\mathcal{O}}(r)$ is nearly independent of the dimension d, and thus the final error scales as $\tilde{\mathcal{O}}(r^{1/2})$.

Remark 4.3. The strategy of appropriately selecting the clipping threshold to ensure that clipping occurs with low probability is commonly applied in the analysis of private algorithms [36, 111]. Adaptive choices of clipping thresholds can provably improve error rates for certain problems including PCA [78] and linear regression [79]. One technical challenge in the choice of the clipping threshold in DPZero is that we need the *expected* one-step progress to be sufficient in Eq. (22). This requires controlling the progress in the low-probability event that finite difference is clipped. The fact that $\|u_t\|$ is finite with probability one simplifies the analysis, which is the reason we choose to sample u_t uniformly at random over the sphere. We believe that the analysis extends to the commonly used spherical Gaussian random vectors, which we leave as a future research direction. Table 7 in the appendix supports our hypothesis that the resulting performances are similar whether Gaussian or spherical random vectors are used. We choose Gaussian vectors for our experiments in Section 5 for simplicity.

Remark 4.4. Our theoretical results, including Theorems 1, 2, and 3, can be extended to the setting where the average loss $F_S(x)$ additionally satisfies the PL inequality [64, 100, 82]. Under Assumption 3.5, DPZero converges to an optimal solution in a nearly dimension-independent error rate. See more details in Appendix F.

Remark 4.5. Per-sample clipping is essential in DP algorithms to ensure bounded sensitivity that determines the magnitude of the DP noise. Besides the dimension-free error rates and memory saving of no backpropagation, another practical merit of DPZero stems from the significantly simplified clipping compared with DP-GD. In addition to the advantage of clipping a *scalar* function value difference rather than a gradient vector as required by first-order methods, the efficiency of DPZero is mainly attributed to the low-cost *per-sample* operations. In DP first-order methods, clipping is applied to gradients for every sample in a batch. The straightforward method of performing backward steps for each sample to compute its gradient loses the benefit of parallelization, leading to significant memory and runtime overhead. Despite extensive effort in improving the efficiency of per-sample gradient clipping [71, 54, 16], these methods still incur extra costs compared to non-DP algorithms. However, the clipping in DPZero only involves computing the per-sample loss from forward steps and incurs no overhead in memory and runtime. This is straightforward for implementation as it is directly supported by, e.g., PyTorch, and no additional techniques are required. DPZero is thus the first private method for fine-tuning LLMs that achieves near-zero additional costs compared to non-DP baselines, which is highly preferable especially in resource-constrained scenarios.

5 Experiments

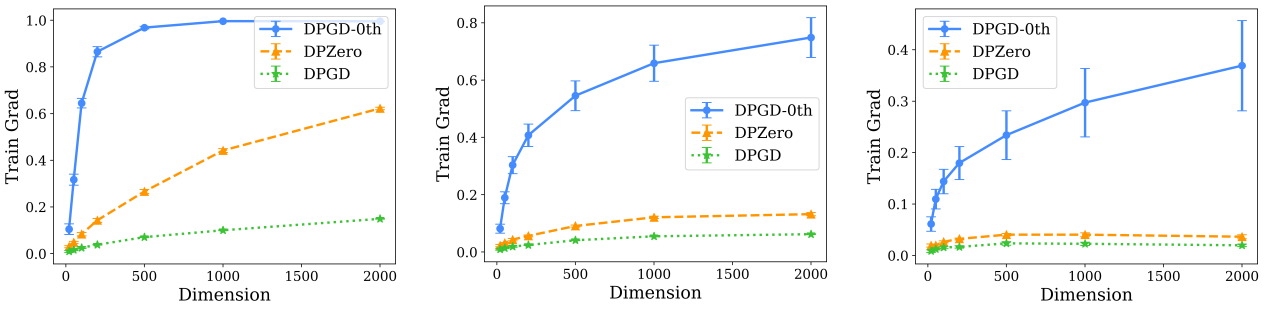
We provide empirical results on synthetic problems and private fine-tuning of language models for sentence classification and generation tasks. A thorough description of the experimental settings is available in Appendix B. All experiments are tested on a single NVIDIA GeForce RTX 3090 GPU with 24 GiB memory. Code is available at `https://github.com/Liang137/DPZero`.

5.1 Synthetic Example

Our first evaluation compares the performance of Algorithm 2 (DPZero) with Algorithm 1 (DPGD-0th) and DP-GD on problems with different effective ranks. In particular, we use a quadratic loss

$$\min_{x\in\mathbb{R}^d} F_S(x) = \frac{1}{2n}\sum_{i=1}^{n}(x-x_i)^\top A(x-x_i),$$

with three choices of the Hessian matrix, A, whose effective ranks are designed to be $\mathcal{O}(d)$, $\mathcal{O}(\sqrt{d})$, and $\mathcal{O}(\log d)$, respectively. All methods are trained with $(\varepsilon = 2, \delta = 10^{-6})$-DP on a training set $\{x_1, \cdots, x_n\}$ with $n = 10,000$ and evaluated on a test set of the same size. The problem dimension is increased from 20 to 2,000. We perform a parameter search and plot the best gradient norm evaluated on both the training set and the test set in Figure 1. Every method scales with the dimension d when the effective rank is d (as in Figures 1(a) and 1(d)), and



(a) $\mathrm{Tr}(A) = \mathcal{O}(d)$. (b) $\mathrm{Tr}(A) = \mathcal{O}(\sqrt{d})$. (c) $\mathrm{Tr}(A) = \mathcal{O}(\log d)$.

(d) $\mathrm{Tr}(A) = \mathcal{O}(d)$. (e) $\mathrm{Tr}(A) = \mathcal{O}(\sqrt{d})$. (f) $\mathrm{Tr}(A) = \mathcal{O}(\log d)$.

Figure 1: Experiments on the quadratic loss with effective rank $\mathrm{Tr}(A)$ (Assumption 3.5). For three different modes of the effective rank, we demonstrate how the norm of the train ((a), (b), and (c)) and test ((d), (e), and (f)) gradient depends on the problem dimension. DPGD-0th (Algorithm 1) has a strong dimension dependence regardless of the effective rank, while DPZero (Algorithm 2) achieves dimension-independent performance when effective rank is small (right panel), similar to the standard first-order method DP-GD. Insights for the saturation of DPGD-0th when the dimension increases can be found in Remark F.5.

DPGD-0th has the worst performance. When the effective rank reduces to $\log d$ (as in Figures 1(c) and 1(f)), both DP-GD and DPZero become nearly dimension-independent, which validates the dimension independence of DPZero. Appendix B.1 includes more results measuring the loss for both training and test datasets.

5.2 Fine-tuning on RoBERTa

Next, we follow the experimental setting in Malladi et al. [87] and evaluate DPZero on fine-tuning RoBERTa [80] with 355M parameters across six different sentence classification tasks. We consider the few-shot scenario with 512 samples per class. We report the test accuracy for DPZero trained with $(\varepsilon = \{2, 6\}, \delta = 10^{-5})$-DP and non-private zeroth-order baseline MeZO [87] and compare them with first-order methods in Table 2. The memory consumption and per-iteration runtime are shown in Table 3. DP first-order methods introduce additional overhead in both memory and runtime compared to non-DP baselines, with a maximum accuracy drop of 9.5% when $\varepsilon = 6$. However, DPZero enjoys the same benefit as MeZO on memory efficiency and achieves near-zero additional costs, with at max only a 2.6% drop in the accuracy. In our experiments, we notice that the clipping threshold of DPZero is typically larger compared to DP first-order methods; see Figure 3 in the appendix. This is consistent with the results in Theorem 3 regarding the selection of the clipping threshold C.

Compared with DP first-order methods, the main benefit of DPZero is memory efficiency. Such memory savings are even greater than those observed in non-DP domains, thanks to DPZero's efficient clipping (cf. Remark 4.5). We note that the aim of Table 3 is to explain that DP first-order methods need considerable memory and runtime overhead compared to non-DP methods, while DPZero does not. Such comparisons happen between DP and non-DP algorithms, respectively. We do not intend to directly compare the runtime of DPZero to DP first-order methods as it depends on the implementation. In general, zeroth-order methods require more iterations to attain the same level of performance as first-order methods [87]. In our case, DP

Table 2: Experiments on RoBERTa (355M). We report both mean and standard error of the accuracy (%) across three random seeds. Zero-shot results with no fine-tuning provide lower bounds (taken from Malladi et al. [87]), since they can be achieved with no private data. MeZO is not private and serves as an upper bound of DPZero. LoRA [58] and DP-LoRA adopt AdamW [83] as their optimizer. All first-order methods (AdamW, LoRA, and their private versions) utilize the implementation by Li et al. [71]. Thanks to DPZero, the performance gaps between zeroth and first-order methods are made smaller in private fine-tuning.

Task	SST-2	SST-5	SNLI	MNLI	RTE	TREC
	Sentiment		Natural Language Inference			Topic
AdamW	93.1 ± 0.3	56.6 ± 0.3	86.4 ± 0.8	81.4 ± 0.9	83.6 ± 1.6	95.9 ± 0.2
DP-AdamW ($\varepsilon = 6$)	91.6 ± 1.2	49.0 ± 0.3	81.5 ± 1.4	76.3 ± 0.9	77.3 ± 1.1	89.9 ± 0.8
DP-AdamW ($\varepsilon = 2$)	90.5 ± 1.5	47.5 ± 0.5	74.6 ± 1.0	70.3 ± 0.8	72.8 ± 0.9	85.0 ± 0.5
LoRA	93.3 ± 0.4	55.3 ± 1.0	85.9 ± 0.7	82.2 ± 0.7	84.2 ± 0.4	94.6 ± 0.4
DP-LoRA ($\varepsilon = 6$)	91.0 ± 1.3	48.8 ± 0.5	81.0 ± 1.5	72.8 ± 1.8	74.7 ± 1.3	89.2 ± 0.8
DP-LoRA ($\varepsilon = 2$)	90.2 ± 1.2	47.1 ± 0.4	74.7 ± 1.6	65.7 ± 0.9	69.2 ± 1.1	83.2 ± 2.3
MeZO	92.5 ± 0.3	50.8 ± 0.8	80.4 ± 0.6	69.2 ± 0.3	72.8 ± 1.0	88.9 ± 0.1
DPZero ($\varepsilon = 6$)	92.2 ± 0.3	49.3 ± 0.6	77.8 ± 1.0	67.4 ± 0.3	71.9 ± 0.9	87.6 ± 0.9
DPZero ($\varepsilon = 2$)	91.8 ± 0.1	47.1 ± 0.9	73.6 ± 0.9	62.7 ± 0.9	70.4 ± 0.7	82.0 ± 1.6
Zero-Shot	79.0	35.5	50.2	48.8	51.4	32.0

Table 3: Runtime per iteration (s) and memory consumption (MiB) when fine-tuning RoBERTa (355M) for SST-2. Private methods in the table ensure ($\varepsilon = 2, \delta = 10^{-5}$)-DP. DPZero is as memory and runtime efficient as the non-private zeroth-order method MeZO [87]. First-order methods DP-AdamW and DP-LoRA (AdamW as the optimizer) both introduce considerable memory and runtime overhead compared to their non-private baselines. All first-order methods use the implementation by Li et al. [71]. Comparisons with other implementations of DP first-order methods can be found in Table 9 in the appendix.

Method	Time (s/iter)	Memory (MiB)
AdamW	1.25	15820
DP-AdamW	2.12	17126
LoRA	0.821	10366
DP-LoRA	1.05	10496
MeZO	0.345	2668
DPZero	0.347	2668

first-order methods take 1,000 iterations while DPZero need 10,000 iterations. This aligns with Theorem 3, which states that DPZero requires $\mathcal{O}(r)$ times more iterations than DP-GD to attain the same level of error rate, where r is the effective rank. However, DPZero can still be efficient for large models in terms of GPU hours, because first-order methods often require communication-heavy distributed training over more GPUs each with limited memory; see Appendix F.6 of Malladi et al. [87].

5.3 Fine-tuning on OPT

We also provide experiments on fine-tuning OPT [148] in the few-shot setting to illustrate the scalability of DPZero. On our device (a GPU with 24 GiB memory), the largest model that can fit in for zeroth-order methods is OPT-6.7B, while first-order methods already run out of memory for OPT-1.3B; see Table 11 in the appendix for a detailed comparison of the memory consumption. The results of DPZero's test performance on four downstream tasks are reported in Tables 4 and 5. DPZero demonstrates the same level of scalability as MeZO, with the ability to fine-tune models wherever MeZO is applicable, and experiences only small drops in performance due to privacy (up to 0.9% when $\varepsilon = 6$). Our results indicate the effectiveness of DPZero for privately fine-tuning pretrained LLMs and confirm that it does not suffer in high dimensions.

Table 4: Experiments on OPT for classification tasks. We report mean and standard error of the accuracy (%) across three random seeds.

Model	OPT-1.3B		OPT-2.7B		OPT-6.7B	
Task	SST-2	BoolQ	SST-2	BoolQ	SST-2	BoolQ
MeZO	88.2 ± 0.9	63.2 ± 0.8	91.9 ± 0.5	65.3 ± 1.3	93.0 ± 0.2	67.4 ± 2.3
DPZero ($\varepsilon = 6$)	88.2 ± 1.1	62.4 ± 0.8	91.5 ± 1.7	65.4 ± 1.6	92.6 ± 0.7	66.8 ± 1.6
DPZero ($\varepsilon = 2$)	86.8 ± 1.7	61.6 ± 1.1	90.5 ± 0.9	63.7 ± 0.7	90.6 ± 1.3	63.7 ± 0.7
Zero-Shot	53.6	45.3	56.3	47.7	61.2	59.4

Table 5: Experiments on OPT for generation tasks. We report both mean and standard error of the f1 score (%) across three random seeds.

Model	OPT-1.3B		OPT-2.7B		OPT-6.7B	
Task	SQuAD	DROP	SQuAD	DROP	SQuAD	DROP
MeZO	73.5 ± 1.2	24.4 ± 0.2	76.3 ± 0.8	25.5 ± 1.2	79.7 ± 1.1	28.8 ± 0.7
DPZero ($\varepsilon = 6$)	72.6 ± 0.8	24.7 ± 1.0	75.7 ± 1.5	24.6 ± 0.5	79.5 ± 0.9	28.4 ± 1.3
DPZero ($\varepsilon = 2$)	70.1 ± 1.6	23.9 ± 1.2	71.9 ± 1.2	23.1 ± 0.9	77.1 ± 1.0	27.6 ± 0.7
Zero-Shot	26.8	11.1	29.8	9.7	36.5	17.8

6 Conclusion

DPZero is proposed to privately fine-tune language models in a memory efficient manner by avoiding backpropagation. Theoretically, DPZero enjoys a provably near dimension-free rate under low-rank structures, clearing the barriers for scaling private fine-tuning of LLMs. When deploying DPZero, the elimination of gradient computation not only significantly saves memory, but avoids the overhead in gradient clipping as well. Thus the benefit of using zeroth-order method is more significant for private optimization. The theoretical guarantees on scalability and the practical merits of DPZero are validated on private fine-tuning of RoBERTa and OPT on several downstream tasks.

DPZero uses the full batch gradient every iteration, and the analysis guarantees an upper bound on the empirical average gradient assuming smooth nonconvex objectives. We defer extensions to the stochastic mini-batch setting, guarantees on the population loss leveraging the stability of zeroth-order methods [95], and considerations of other assumptions on objective functions like convexity or nonsmoothness to future research. We believe this work opens up a plethora of other prospective directions in DP zeroth-order optimization. These include, but are not limited to, understanding advantages of the intrinsic noise in zeroth-order gradient estimators, discovering other structural assumptions like the restricted Lipschitz condition [70] for dimension-independent rates, exploring alternative private mechanisms for the privacy guarantees of DPZero (e.g., the Laplace mechanism for pure DP [117]), and utilizing momentum [122] or variance reduction [4] techniques for an improved rate and computational complexity.

Acknowledgements

We are grateful to Gavin Brown and Divyansh Pareek for their insightful discussions regarding the proofs. We also thank Fanny Yang for proofreading of the paper. Additionally, we thank all anonymous reviewers for their valuable suggestions. L.Z. gratefully acknowledges funding by the Max Planck ETH Center for Learning Systems (CLS). This work does not relate to the current position of K.T. at Amazon. N.H. is supported by ETH research grant funded through ETH Zurich Foundations and Swiss National Science Foundation Project Funding No. 200021-207343. S.O. is supported in part by the National Science Foundation under grant no. 2019844, 2112471, and 2229876 supported in part by funds provided by the National Science Foundation, by the Department of Homeland Security, and by IBM. Any opinions, findings, and conclusions or recommendations

expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or its federal agency and industry partners.

Impact Statement

A major concern with current use-cases of large language models is privacy of the fine-tuning data. Fine-tuning on in-domain data greatly improves performance and is now a default option. However, in-domain data can contain sensitive information about the participants of the dataset. The proposed solution makes privacy protection easier, consuming less resources, thus democratizing the use of privacy enhancing technology beyond those who have access to large amounts of resources.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 7319–7328, 2021.

[3] Rohan Anil, Vineet Gupta, Tomer Koren, and Yoram Singer. Memory efficient adaptive optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[4] 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.

[5] Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: Optimal rates in ℓ_1 geometry. In *International Conference on Machine Learning*, pages 393–403. PMLR, 2021.

[6] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

[7] Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. *Advances in Neural Information Processing Systems*, 31, 2018.

[8] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *IEEE Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[9] 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.

[10] Raef Bassily, Vitaly Feldman, Cristóbal Guzmán, and Kunal Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33, 2020.

[11] Atılım Güneş Baydin, Barak A Pearlmutter, Don Syme, Frank Wood, and Philip Torr. Gradients without backpropagation. *arXiv preprint arXiv:2202.08587*, 2022.

[12] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge, 2009.

[13] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

[14] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[15] Zhiqi Bu, Justin Chiu, Ruixuan Liu, Sheng Zha, and George Karypis. Zero redundancy distributed learning with differential privacy. *arXiv preprint arXiv:2311.11822*, 2023.

[16] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Differentially private optimization on large model at small cost. In *International Conference on Machine Learning*, pages 3192–3218. PMLR, 2023.

[17] HanQin Cai, Daniel Mckenzie, Wotao Yin, and Zhenliang Zhang. Zeroth-order regularized optimization (ZORO): Approximately sparse gradients and adaptive sampling. *SIAM Journal on Optimization*, 32(2): 687–714, 2022.

[18] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[19] Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Konstantinos Parasyris, Jiancheng Liu, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. DeepZero: Scaling up zeroth-order optimization for deep model training. In *International Conference on Learning Representations*, 2024.

[20] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017.

[21] Tiejin Chen, Longchao Da, Huixue Zhou, Pingzhi Li, Kaixiong Zhou, Tianlong Chen, and Hua Wei. Privacy-preserving fine-tuning of large language models through flatness. *arXiv preprint arXiv:2403.04124*, 2024.

[22] Xiangyi Chen, Sijia Liu, Kaidi Xu, Xingguo Li, Xue Lin, Mingyi Hong, and David Cox. ZO-AdaMM: Zeroth-order adaptive momentum method for black-box optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[23] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private SGD: A geometric perspective. *Advances in Neural Information Processing Systems*, 33:13773–13782, 2020.

[24] Krzysztof Choromanski, Mark Rowland, Vikas Sindhwani, Richard Turner, and Adrian Weller. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pages 970–978. PMLR, 2018.

[25] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2924–2936, 2019.

[26] Harald Cramér. *Mathematical methods of statistics*, volume 43. Princeton University Press, 1999.

[27] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge, 2005.

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

[29] Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. DP-Forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 2665–2679, 2023.

[30] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2368–2378, 2019.

[31] John C Duchi, Peter L Bartlett, and Martin J Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[32] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

[33] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[34] 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.

[35] 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.

[36] Huang Fang, Xiaoyun Li, Chenglin Fan, and Ping Li. Improved convergence of differential private SGD with gradient clipping. In *International Conference on Learning Representations*, 2023.

[37] Wenzhi Fang, Ziyi Yu, Yuning Jiang, Yuanming Shi, Colin N Jones, and Yong Zhou. Communication-efficient stochastic zeroth-order optimization for federated learning. *IEEE Transactions on Signal Processing*, 70:5058–5073, 2022.

[38] 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.

[39] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 385–394, 2005.

[40] Arun Ganesh, Mahdi Haghifam, Milad Nasr, Sewoong Oh, Thomas Steinke, Om Thakkar, Abhradeep Guha Thakurta, and Lun Wang. Why is public pretraining necessary for private model training? In *International Conference on Machine Learning*, pages 10611–10627. PMLR, 2023.

[41] Arun Ganesh, Mahdi Haghifam, Thomas Steinke, and Abhradeep Guha Thakurta. Faster differentially private convex optimization via second-order methods. *Advances in Neural Information Processing Systems*, 36, 2023.

[42] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 3816–3830, 2021.

[43] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[44] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via Hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019.

[45] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. The third PASCAL recognizing textual entailment challenge, 2007.

[46] Daniel Golovin, John Karro, Greg Kochanski, Chansoo Lee, Xingyou Song, and Qiuyi Zhang. Gradientless descent: High-dimensional zeroth-order optimization. In *International Conference on Learning Representations*, 2020.

[47] Cristiano Gratton, Naveen KD Venkategowda, Reza Arablouei, and Stefan Werner. Privacy-preserved distributed learning with zeroth-order optimization. *IEEE Transactions on Information Forensics and Security*, 17:265–279, 2021.

[48] Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. *Advances in Neural Information Processing Systems*, 28, 2015.

[49] Abhradeep Guha Thakurta and Adam Smith. (Nearly) optimal algorithms for private online learning in full-information and bandit settings. *Advances in Neural Information Processing Systems*, 26, 2013.

[50] Arjun K Gupta and Saralees Nadarajah. *Handbook of Beta distribution and its applications*. CRC Press, 2004.

[51] Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

[52] R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second PASCAL recognising textual entailment challenge, 2006.

[53] Andi Han, Bamdev Mishra, Pratik Jawanpuria, and Junbin Gao. Differentially private Riemannian optimization. *Machine Learning*, 113(3):1133–1161, 2024.

[54] Jiyan He, Xuechen Li, Da Yu, Huishuai Zhang, Janardhan Kulkarni, Yin Tat Lee, Arturs Backurs, Nenghai Yu, and Jiang Bian. Exploring the limits of differentially private deep learning with group-wise clipping. In *International Conference on Learning Representations*, 2023.

[55] Geoffrey Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022.

[56] Junyuan Hong, Jiachen T. Wang, Chenhui Zhang, Zhangheng LI, Bo Li, and Zhangyang Wang. DP-OPT: Make large language model your privacy-preserving prompt engineer. In *International Conference on Learning Representations*, 2024.

[57] Bairu Hou, Joe O'connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. PromptBoosting: Black-box text classification with ten forward passes. In *International Conference on Machine Learning*, pages 13309–13324. PMLR, 2023.

[58] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[59] Zonghao Huang, Rui Hu, Yuanxiong Guo, Eric Chan-Tin, and Yanmin Gong. DP-ADMM: ADMM-based distributed learning with differential privacy. *IEEE Transactions on Information Forensics and Security*, 15:1002–1012, 2019.

[60] Prateek Jain and Abhradeep Guha Thakurta. (Near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, pages 476–484. PMLR, 2014.

[61] Kaiyi Ji, Zhe Wang, Yi Zhou, and Yingbin Liang. Improved zeroth-order variance reduced algorithms and analysis for nonconvex optimization. In *International Conference on Machine Learning*, pages 3100–3109. PMLR, 2019.

[62] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International Conference on Machine Learning*, pages 1376–1385. PMLR, 2015.

[63] Peter Kairouz, Monica Ribero Diaz, Keith Rush, and Abhradeep Thakurta. (Nearly) dimension independent private ERM with adagrad rates via publicly estimated subspaces. In *Conference on Learning Theory*, pages 2717–2746. PMLR, 2021.

[64] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811, 2016.

[65] Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. Privacy via the Johnson-Lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5(1):39–71, 2013.

[66] Anastasia Koloskova, Hadrien Hendrikx, and Sebastian U Stich. Revisiting gradient clipping: Stochastic bias and tight convergence guarantees. In *International Conference on Machine Learning*, 2023.

[67] Janardhan Kulkarni, Yin Tat Lee, and Daogao Liu. Private non-smooth ERM and SCO in subquadratic steps. *Advances in Neural Information Processing Systems*, 34, 2021.

[68] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

[69] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pages 4582–4597, 2021.

[70] Xuechen Li, Daogao Liu, Tatsunori B Hashimoto, Huseyin A Inan, Janardhan Kulkarni, Yin-Tat Lee, and Abhradeep Guha Thakurta. When does differentially private learning not suffer in high dimensions? *Advances in Neural Information Processing Systems*, 35:28616–28630, 2022.

[71] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022.

[72] Xiangru Lian, Huan Zhang, Cho-Jui Hsieh, Yijun Huang, and Ji Liu. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. *Advances in Neural Information Processing Systems*, 29, 2016.

[73] Tianyi Lin, Zeyu Zheng, and Michael Jordan. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 35:26160–26175, 2022.

[74] 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, 2023.

[75] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[76] Sijia Liu, Pin-Yu Chen, Xiangyi Chen, and Mingyi Hong. SignSGD via zeroth-order oracle. In *International Conference on Learning Representations*, 2019.

[77] Terrance Liu, Jingwu Tang, Giuseppe Vietri, and Steven Wu. Generating private synthetic data with genetic algorithms. In *International Conference on Machine Learning*, pages 22009–22027. PMLR, 2023.

[78] Xiyang Liu, Weihao Kong, Prateek Jain, and Sewoong Oh. DP-PCA: Statistically optimal and differentially private PCA. *Advances in Neural Information Processing Systems*, 35:29929–29943, 2022.

[79] Xiyang Liu, Prateek Jain, Weihao Kong, Sewoong Oh, and Arun Suggala. Label robust and differentially private linear regression: Computational and statistical efficiency. *Advances in Neural Information Processing Systems*, 36, 2023.

[80] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[81] Zhihao Liu, Jian Lou, Wenjie Bao, Yuke Hu, Bo Li, Zhan Qin, and Kui Ren. Differentially private zeroth-order methods for scalable large language model finetuning. *arXiv preprint arXiv:2402.07818*, 2024.

[82] Stanislaw Łojasiewicz. A topological property of real analytic subsets. *Coll. du CNRS, Les équations aux dérivées partielles*, 117(87-89):2, 1963.

[83] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[84] Andrew Lowy, Zeman Li, Tianjian Huang, and Meisam Razaviyayn. Optimal differentially private learning with public data. *arXiv preprint arXiv:2306.15056*, 2023.

[85] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *IEEE Symposium on Security and Privacy*, pages 346–363. IEEE, 2023.

[86] Yi-An Ma, Teodor Vanislavov Marinov, and Tong Zhang. Dimension independent generalization of DP-SGD for overparameterized smooth convex optimization. *arXiv preprint arXiv:2206.01836*, 2022.

[87] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

[88] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[89] George Marsaglia. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, 43(2):645–646, 1972.

[90] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*, 2023.

[91] Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David Evans, and Taylor Berg-Kirkpatrick. Memorization in NLP fine-tuning methods. *arXiv preprint arXiv:2205.12506*, 2022.

[92] Mervin E Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2(4):19–20, 1959.

[93] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[94] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

[95] Konstantinos Nikolakakis, Farzin Haddadpour, Dionysis Kalogerias, and Amin Karbasi. Black-box generalization: Stability of zeroth-order learning. *Advances in Neural Information Processing Systems*, 35:31525–31541, 2022.

[96] OpenAI. GPT-4 Technical Report, 2023.

[97] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[98] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[99] Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. HyperTuning: Toward adapting large language models without back-propagation. In *International Conference on Machine Learning*, pages 27854–27875. PMLR, 2023.

[100] Boris T Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

[101] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI*, 2018.

[102] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

[103] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[104] Matthew Reimherr, Karthik Bharath, and Carlos Soto. Differential privacy over Riemannian manifolds. *Advances in Neural Information Processing Systems*, 34:12292–12303, 2021.

[105] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the Hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

[106] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

[107] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

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

[109] Roshan Shariff and Or Sheffet. Differentially private contextual linear bandits. *Advances in Neural Information Processing Systems*, 31, 2018.

[110] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR, 2018.

[111] Zebang Shen, Jiayuan Ye, Anmin Kang, Hamed Hassani, and Reza Shokri. Share your representation only: Guaranteed improvement of the privacy-utility tradeoff in federated learning. In *International Conference on Learning Representations*, 2023.

[112] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, pages 3–18. IEEE, 2017.

[113] David Silver, Anirudh Goyal, Ivo Danihelka, Matteo Hessel, and Hado van Hasselt. Learning by directional gradient descent. In *International Conference on Learning Representations*, 2022.

[114] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[115] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

[116] Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private GLMs. In *International Conference on Artificial Intelligence and Statistics*, pages 2638–2646. PMLR, 2021.

[117] Xinyu Tang, Ashwinee Panda, Milad Nasr, Saeed Mahloujifar, and Prateek Mittal. Private fine-tuning of large language models with zeroth-order optimization. *arXiv preprint arXiv:2401.04343*, 2024.

[118] Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. Privacy-preserving in-context learning with differentially private few-shot generation. In *International Conference on Learning Representations*, 2024.

[119] Aristide Tossou and Christos Dimitrakakis. Algorithms for differentially private multi-armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[120] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[121] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. LLAMA 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[122] Hoang Tran and Ashok Cutkosky. Momentum aggregation for private non-convex ERM. *Advances in Neural Information Processing Systems*, 35:10996–11008, 2022.

[123] Saiteja Utpala, Andi Han, Pratik Jawanpuria, and Bamdev Mishra. Improved differentially private Riemannian optimization: Fast sampling and variance reduction. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[124] Saiteja Utpala, Praneeth Vepakomma, and Nina Miolane. Differentially private Fréchet mean on the manifold of symmetric positive definite (SPD) matrices with log-Euclidean metric. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[125] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

[126] Paul Voigt and Axel Von dem Bussche. The EU general data protection regulation (GDPR). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[127] Ellen M Voorhees and Dawn M Tice. Building a question answering test collection. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 200–207, 2000.

[128] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[129] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

[130] 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.

[131] 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.

[132] Yining Wang, Simon Du, Sivaraman Balakrishnan, and Aarti Singh. Stochastic zeroth-order optimization in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pages 1356–1365. PMLR, 2018.

[133] Zhongruo Wang, Krishnakumar Balasubramanian, Shiqian Ma, and Meisam Razaviyayn. Zeroth-order algorithms for nonconvex–strongly-concave minimax problems with improved complexities. *Journal of Global Optimization*, pages 1–32, 2022.

[134] Andre Wibisono, Martin J Wainwright, Michael Jordan, and John C Duchi. Finite sample convergence rates of zero-order stochastic optimization methods. *Advances in Neural Information Processing Systems*, 25, 2012.

[135] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1112–1122, 2018.

[136] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of ACM International Conference on Management of Data*, pages 1307–1322, 2017.

[137] Mengwei Xu, Yaozong Wu, Dongqi Cai, Xiang Li, and Shangguang Wang. Federated fine-tuning of billion-sized language models across mobile devices. *arXiv preprint arXiv:2308.13894*, 2023.

[138] Xiaodong Yang, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Normalized/Clipped SGD with perturbation for differentially private non-convex optimization. *arXiv preprint arXiv:2206.13033*, 2022.

[139] Farzad Yousefian, Angelia Nedić, and Uday V Shanbhag. On stochastic gradient and subgradient methods with adaptive steplength sequences. *Automatica*, 48(1):56–67, 2012.

[140] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*, 2022.

[141] Pengyun Yue, Long Yang, Cong Fang, and Zhouchen Lin. Zeroth-order optimization with weak dimension dependency. In *Annual Conference on Learning Theory*, pages 4429–4472. PMLR, 2023.

[142] Eric Zelikman, Qian Huang, Percy Liang, Nick Haber, and Noah D Goodman. Just one byte (per gradient): A note on low-bandwidth decentralized language model finetuning using shared randomness. *arXiv preprint arXiv:2306.10015*, 2023.

[143] Shenglai Zeng, Yaxin Li, Jie Ren, Yiding Liu, Han Xu, Pengfei He, Yue Xing, Shuaiqiang Wang, Jiliang Tang, and Dawei Yin. Exploring memorization in fine-tuned language models. *arXiv preprint arXiv:2310.06714*, 2023.

[144] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private ERM for smooth objectives. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3922–3928, 2017.

[145] Liang Zhang, Kiran K Thekumparampil, Sewoong Oh, and Niao He. Bring your own algorithm for optimal differentially private stochastic minimax optimization. *Advances in Neural Information Processing Systems*, 35:35174–35187, 2022.

[146] Liang Zhang, Kiran K Thekumparampil, Sewoong Oh, and Niao He. DPZero: Dimension-independent and differentially private zeroth-order optimization. *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS*, 2023.

[147] Qinzi Zhang, Hoang Tran, and Ashok Cutkosky. Private zeroth-order nonsmooth nonconvex optimization. In *International Conference on Learning Representations*, 2024.

[148] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained Transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[149] Xinwei Zhang, Zhiqi Bu, Steven Wu, and Mingyi Hong. Differentially private SGD without clipping bias: An error-feedback approach. In *International Conference on Learning Representations*, 2024.

[150] 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.

[151] Yingxue Zhou, Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private SGD with gradient subspace identification. In *International Conference on Learning Representations*, 2021.

A Additional Related Works

Zeroth-order optimization. Nesterov and Spokoiny [94] pioneered the formal analysis of the convergence rate of zeroth-order methods, i.e., zeroth-order (stochastic) gradient descent (ZO-SGD) that replaces gradients in SGD by their zeroth-order estimators. This is motivated by renewed interest in adopting zeroth-order methods in industry due to, for example, fast differentiation techniques that require storing all intermediate computations reaching the memory limitations. Their findings on nonsmooth convex functions are later refined by Shamir [108]. Lin et al. [73] contributed to further advancements on nonsmooth nonconvex functions recently. Additionally, Ghadimi and Lan [43] extended the results for smooth functions into the stochastic setting. Zeroth-order methods have also been expanded to incorporate approaches such as coordinate descent [72], conditional gradient descent [7], variance reduction techniques [75, 35, 61], SignSGD [76], and minimax optimization [133]. Additionally, zeroth-order methods find applications in fields such as black-box machine learning [48, 20, 22], bandit optimization [39, 108], reinforcement learning [106, 24, 88], and distributed learning [37, 142, 137] to reduce communication overhead.

These well-established results indicate that the norm of the zeroth-order gradient scales with the dimension d and the required stepsize is d-times smaller than that in first-order gradient-based methods, leading to a d-times increase in the final time complexity. For example, the convergence rate of gradient descent for minimizing a smooth convex function $f(x)$ is $f(\bar{x}_T) - \min_{x\in\mathbb{R}^d} f(x) \leq \mathcal{O}(1/T)$ where $\bar{x}_T$ is the average of T iterates [93], while the zeroth-order method only achieves a rate $\mathcal{O}(d/T)$. It has been shown that such dimension dependence of zeroth-order methods is inevitable without additional structures [134, 32].

There are several recent works that relax the dimension dependence in zeroth-order methods leveraging problem structures. Wang et al. [132] and Cai et al. [17] assumed certain sparsity structure in the problem and applied sparse recovering algorithms, e.g. LASSO, to obtain sparse gradients from zeroth-order observations. Golovin et al. [46] analyzed the case when the objective function is $f(Px)$ for some low-rank projection matrix P. These works either require the objective or the algorithm to be modified to have a dimension-independent guarantee. Balasubramanian and Ghadimi [7] demonstrated that ZO-SGD can directly identify the sparsity of the problem and proved a dimension-independent rate when the support of gradients remains unchanged [17]. Recently, Yue et al. [141] and Malladi et al. [87] relaxed the dependence on dimension d to a quantity related to the trace of the loss's Hessian.

Differentially private optimization. Previous works on DP optimization mostly center around first-order methods. For constrained convex problems, tight utility guarantees on both excess empirical [18, 8, 136, 144, 130] and population [9, 10, 38, 5, 67, 145] losses are well-understood. As an example, a typical result states that the optimal rate on the excess empirical loss for convex objectives is $\Theta(\sqrt{d\log(1/\delta)}/(n\varepsilon))$, where (ε, δ) are privacy parameters, n is the number of samples, and d is the dimension. The dimension dependence is fundamental as both the upper bound [8], using differentially private (stochastic) gradient descent (DP-GD) introduced in [115], and the lower bound [8], using a reduction to finger printing codes, have the same dependence.

When the problem is nonconvex, i.e., the setting of our interest, DP-GD achieves a rate of $\mathcal{O}(\sqrt{d\log(1/\delta)}/(n\varepsilon))$ on the squared norm of the gradient [130, 150]. We show that DPZero matches this rate with access only to the zeroth-order oracle in Theorem 3. Given access to the first-order oracle, it has been recently shown that such rate can be improved to $\mathcal{O}((\sqrt{d\log(1/\delta)}/(n\varepsilon))^{4/3})$ leveraging momentum [122] or variance reduction techniques [4]. Further, the convergence to second-order stationary points in nonconvex DP optimization is studied in [74]. Recent advancements in DP optimization have also delved into the understanding of the potential of public data [40, 84], the convergence properties of per-sample gradient clipping [138, 36, 66, 149], and the relaxation of the dimension dependence in the utility upper bound [86, 70].

Early works established that dimension-independent rates can be attained when the gradients lie in some fixed low-rank subspace [60, 116]. By first identifying this gradient subspace, dimension-independent algorithms can be designed [151, 63]. Closest to our result is Song et al. [116], which demonstrated that the rate of DP-GD for smooth nonconvex optimization can be improved to $\mathcal{O}(\sqrt{r\log(1/\delta)}/(n\varepsilon))$ under certain structural assumptions, i.e., for generalized linear models (GLMs) with a rank-r feature matrix. DPZero matches this result with access only to the zeroth-order oracle in Theorem 3 for more general problems beyond low-rank GLMs. Our result is inspired by Li et al. [70] that introduced a relaxed Lipschitz condition for the gradients and provided dimension-free bounds when the loss is convex and the relaxed Lipschitz parameters decay rapidly. Similarly, Ma et al. [86] suggested that the dependence on d in the utility upper bound for DP stochastic

convex optimization can be improved to a dependence on the trace of the Hessian. There is also a line of work on DP Riemannian optimization that achieves utility bounds dependent on the intrinsic dimension of the manifold [104, 124, 123, 53]. Further exploration of its connection to the low-rank structure in this work is reserved for future.

Literature on DP optimization beyond first-order methods remains less explored. Ganesh et al. [41] investigated the potential of second-order methods for DP convex optimization. Gratton et al. [47] proposed to use zeroth-order methods for DP-ADMM [59] in distributed learning. They state that the noise intrinsic in zeroth-order methods is enough to provide privacy guarantee and rely on the output of zeroth-order methods being Gaussian, which is unverified to the best of our knowledge. Liu et al. [77] proposed a private genetic algorithm based on zeroth-order optimization heuristics for private synthetic data generation. Recently, Zhang et al. [147] studied the problem of private zeroth-order nonsmooth nonconvex optimization and achieved a rate that depends on the dimension d. After the workshop version of our paper [146] was released, Tang et al. [117] concurrently discovered the same algorithm as DPZero (up to a minor difference in how u_t is drawn) and showed empirical benefits when applied to fine-tuning OPT models but without theoretical analysis. Also building upon the workshop version of our paper, Liu et al. [81] introduced DP-ZOSO, a stage-wise zeroth-order method with an additional quadratic regularizer. With extra hyper-parameters to be tuned, DP-ZOSO demonstrates further empirical gain over DPZero. However, Liu et al. [81] only provided dimension-dependent guarantees. As far as we are aware, no prior studies have addressed the challenge of deriving a dimension-independent rate in DP zeroth-order optimization.

Other relevant works. Du et al. [29] introduced a novel noise adding mechanism that happens in the forward pass of training. Although the algorithm is termed "DP-Forward", it still requires backpropagation for training. In a separate context, Bu et al. [15] coincidentally proposed DP-ZeRO, a term identical to ours, denoting a private version of the zero redundancy optimizer (ZeRO) by Rajbhandari et al. [102] that aims at enhancing memory efficiency in data and model parallelisms. DP prompt tuning [56] and DP in-context learning [118] provide resource-efficient alternatives compared to private fine-tuning, enabling the private adaptation of pretrained LLMs to specific tasks without extensive computational demands. Investigating how DPZero performs relative to these methods and whether different techniques can be integrated is an interesting research problem. More recently, Chen et al. [21] proposed differentially private algorithms that enforce weight flatness to improve generalization, which can also handle zeroth-order oracles. There is also another line of research [49, 119, 109] on the design of differentially private algorithms for the stochastic bandit problem based on upper confidence bound [6]. Their algorithms are not directly applicable to our setting.

B Additional Experiment Details

In this section, we discuss our experimental setups in detail.

B.1 Synthetic Example on a Quadratic Loss

Given a training dataset $S = \{x_1, \cdots, x_n\}$ with each coordinate of $x_i \in \mathbb{R}^d$ sampled independently from the Gaussian $\mathcal{N}(1, 1)$, we implement DPZero on the quadratic loss

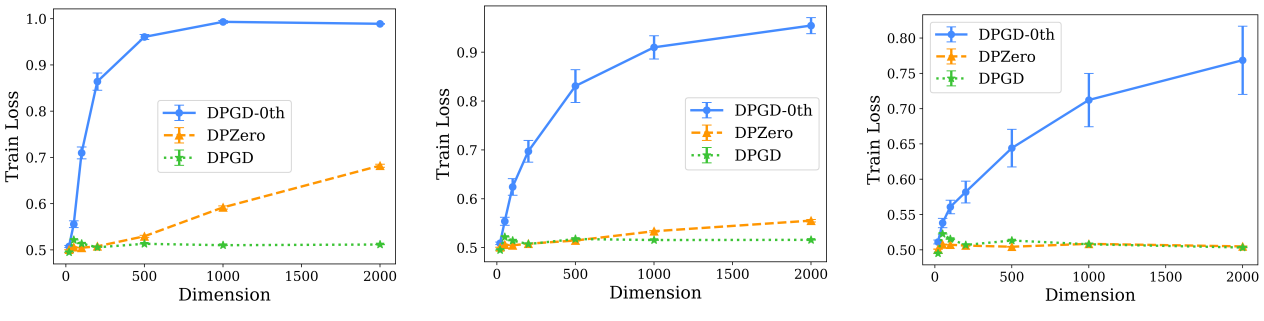
$$\min_{x \in \mathbb{R}^d} F_S(x) = \frac{1}{2n} \sum_{i=1}^{n} (x - x_i)^\top A (x - x_i),$$

with a fixed Hessian $A \in \mathbb{R}^{d \times d}$ that can be designed to implement different effective ranks $r = \mathrm{Tr}(A)/\|A\|_2$ according to Assumption 3.5. We compare DPZero (Algorithm 2) with DPGD-0th (Algorithm 1) and first-order algorithm DP-GD on three patterns of the effective rank

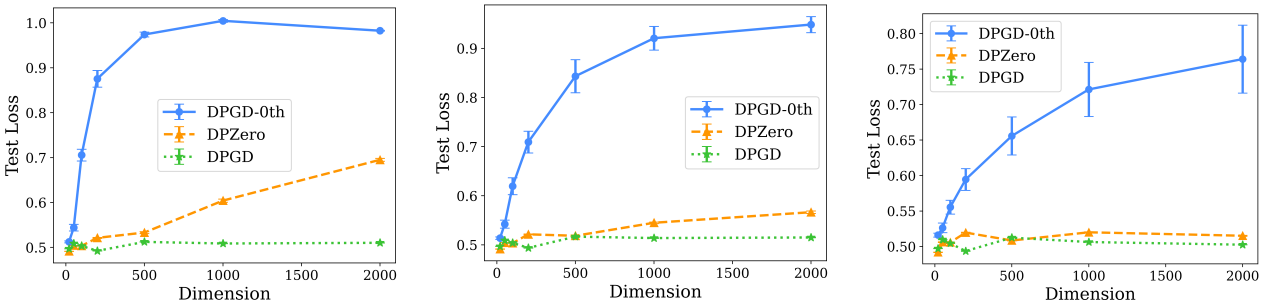
(a) $\mathrm{Tr}(A) = \mathcal{O}(d)$: $A = \mathrm{diag}\{1, 1, \cdots, 1\}$;

(b) $\mathrm{Tr}(A) = \mathcal{O}(\sqrt{d})$: $A = \mathrm{diag}\{1, 1/\sqrt{2}, \cdots, 1/\sqrt{d}\}$;

(c) $\mathrm{Tr}(A) = \mathcal{O}(\log d)$: $A = \mathrm{diag}\{1, 1/2, \cdots, 1/d\}$.



(a) $\mathrm{Tr}(A) = \mathcal{O}(d)$. (b) $\mathrm{Tr}(A) = \mathcal{O}(\sqrt{d})$. (c) $\mathrm{Tr}(A) = \mathcal{O}(\log d)$.



(d) $\mathrm{Tr}(A) = \mathcal{O}(d)$. (e) $\mathrm{Tr}(A) = \mathcal{O}(\sqrt{d})$. (f) $\mathrm{Tr}(A) = \mathcal{O}(\log d)$.

Figure 2: Experiments on the quadratic loss with effective rank $\mathrm{Tr}(A)$. For three different modes, we increase the dimension and report the best loss evaluated on both training set ((a), (b), and (c)) and test set ((d), (e), and (f)).

Since $\|A\|_2 = 1$ in all cases, the effective rank $r = \mathrm{Tr}(A)$. For each mode of the effective rank, we increase the problem dimension d from 20 to 2000. We perform a parameter search and plot the best gradient norm evaluated on the training set and a test set that follows the same distribution of the training set in Figure 1. For completeness, we also plot both training and test loss in Figure 2. The key hyper-parameters used for the experiments are summarized in Table 6.

Table 6: Hyper-parameters used for the synthetic example on the quadratic loss. The number of iterations, stepsize, and clipping threshold are optimized through a grid search using given values. Other parameters are fixed to the values.

Hyper-parameters	Values
Number of training samples	10000
Number of test samples	10000
Dimension d	$\{20, 50, 100, 200, 500, 1000, 2000\}$
Privacy	$(\varepsilon = 2, \delta = 10^{-6})$
Smoothing λ (DPZero and DPGD-0th)	10^{-4}
Number of iterations	$\{10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120\}$
Stepsize	$\{10^{-5}, 3\times10^{-5}, 10^{-4}, 3\times10^{-4}, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1\}$
Clipping	$\{0.1, 0.3, 1, 3, 10, 30, 100, 300\}$

In all figures, we observe that the performance of each method is improved with smaller effective rank. For each pattern of the effective rank, DPGD-0th (Algorithm 1) has the worst performance, while DP-GD consistently achieves the best results. When the effective rank is d, every method scales with the dimension. When the effective rank improves to $\log d$, DPZero and DP-GD become nearly dimension-independent, and DPZero matches the performance of the first-order method DP-GD. This validates our theoretical findings, as summarized in Table 1, and demonstrates the effectiveness of DPZero. We want to mention that a similar set of experiments to verify the performance of DP-GD when dimension increases was also provided by Li et al. [70]. Our implementation of this synthetic example is based on their code.

B.2 Private Fine-Tuning of the Language Model RoBERTa

We follow experiment settings in Malladi et al. [87] to evaluate the performance of DPZero in the private fine-tuning of RoBERTa [80] across six sentence classification datasets: SST-2 and SST-5 [114] for sentiment classification, SNLI [13], MNLI [135], and RTE [27, 52, 45, 12, 129] for natural language inference tasks, and TREC [127] for topic classification. In our experiments, we employ the same prompts as used in Malladi et al. [87], which are adapted from Gao et al. [42].

Implementation details. Our implementation of DPZero utilizes the codebase provided by Malladi et al. [87]. For easier implementation and better memory efficiency, we follow Malladi et al. [87] to sample the zeroth-order direction u_t from the Gaussian distribution $\mathcal{N}(0, \mathrm{I}_d)$ instead of the sphere as stated in Algorithm 2. Table 7 compares the performance of DPZero on SST-2 and SST-5 when u_t is sampled from Gaussian and sphere. Given the negligible differences between the two sampling strategies, we continue with the Gaussian sampling for its simplicity. Another strategy in the implementation to further save memory involves storing only the random seed for the generation of the zeroth-order direction u_t, rather than the complete vector, and regenerating this direction whenever it's used. Although DPZero is stated for the full-batch case in Algorithm 2, we adopt a mini-batch setting in the experiments.

Table 7: Test accuracy (mean % ± standard error %) of DPZero when fine-tuning RoBERTa (355M) for SST-2 and SST-5 with ($\varepsilon = \{2, 6\}, \delta = 10^{-5}$)-DP and using different sampling strategies of the zeroth-order update direction u_t. No notable difference is observed when u_t is sampled from either the Gaussian distribution or the Euclidean sphere.

Randomness	Gaussian		Sphere	
	$\varepsilon = 6$	$\varepsilon = 2$	$\varepsilon = 6$	$\varepsilon = 2$
SST-2	92.2 ± 0.3	91.8 ± 0.1	91.8 ± 0.1	91.5 ± 0.5
SST-5	49.3 ± 0.6	47.1 ± 0.9	49.9 ± 1.3	47.4 ± 1.3

Hyper-parameter selection. For all experiments, we employ a few-shot setting, utilizing 512 samples per class in the training set, randomly selected from the original dataset. The test set is also composed of 1000 randomly selected samples from the original test dataset. We fix the total number of iterations to be 10000, the batch size to be 64, and the smoothing parameter $\lambda = 10^{-3}$ for both DPZero and the non-private zeroth-order baseline MeZO [87]. Note that the original results of MeZO reported in Malladi et al. [87] run for 100000 iterations. A parameter search of the learning rate for MeZO is performed, and it turns out 10^{-6} consistently yields the best performance. We then fix the learning rate to be 10^{-6} for DPZero and only search for the clipping threshold for different tasks. There is potential for improved performance by well-optimizing other hyper-parameters, such as the learning rate and the number of iterations. All results are averaged through three different random seeds $\{42, 13, 21\}$ for selecting the few-shot datasets. The hyper-parameters used for our language model fine-tuning experiments are summarized in Table 8.

Comparison with first-order methods. Regarding the first-order methods, we use the same few-shot setting as before, and the results are averaged over three different random seeds $\{42, 13, 21\}$. The number of iterations is set to be 1000, and the batch size is fixed to be 64. The learning rate is optimized by a grid search over $\{5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$, and the clipping threshold is optimized by a grid search over $\{0.1, 0.5, 1, 10\}$. In the experiments for LoRA, we set the rank to be 8 and the LoRA $\alpha = 16$, which remain the same as in the original paper [58]. All other parameters are fixed to their default values. In addition to Li et al. [71] in Tables 2 and 3, we also compare the performance of DPZero to two other implementations of DP first-order methods, Yu et al. [140] and Bu et al. [16], in Table 9. DPZero achieves similar performance on SST-2 as DP first-order methods, while saving a significant amount of memory. Such memory savings are greater than the savings of MeZO [87] over AdamW [83] and LoRA [58] (AdamW as the optimizer), due to DPZero's simpler clipping (cf. Remark 4.5).

Table 8: Hyper-parameters used in DPZero for fine-tuning RoBERTa (355M). We only optimize the clipping threshold through a grid search from 50 to 400. Other parameters are fixed to the listed values.

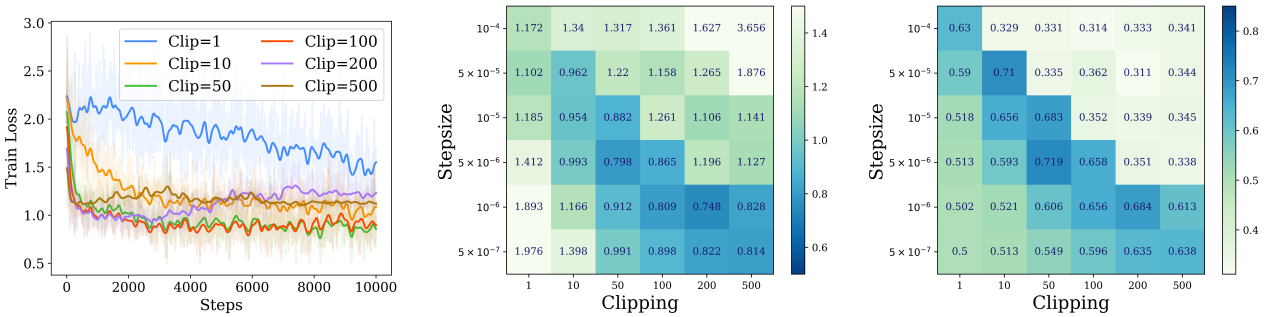
Hyper-parameters	Values
Number of training samples	512 per class
Number of test samples	1000
Number of iterations	10000
Batch size	64
Privacy	$(\varepsilon = \{2, 6\}, \delta = 10^{-5})$
Smoothing λ	10^{-3}
Stepsize	10^{-6}
Clipping	{50, 100, 150, 200, 250, 300, 400}

Table 9: Test accuracy (%), runtime per iteration (s), and memory consumption (MiB) when fine-tuning RoBERTa (355M) for SST-2. Private methods in the table guarantee $(\varepsilon = 2, \delta = 10^{-5})$-DP. A fair comparison is ensured among Li et al. [71] and Bu et al. [16], as they are implemented using the same codebase. It is important to note, however, that they cannot be directly compared with those of Yu et al. [140], due to differences in implementations. LoRA [58] and DP-LoRA use the first-order method AdamW [83] as the optimizer. DP first-order methods introduce considerable overheads in both memory and runtime compared to their non-DP baselines, while DPZero does not, thanks to its novel design of the efficient clipping. Also note that such comparisons between DP and non-DP algorithms are fair since they use the same codebase.

Method	Acc.	Time (s/iter)	Memory (MiB)
AdamW [71]	93.1	1.25	15820
DP-AdamW [71]	90.5	2.12	17126
DP-AdamW [16]	91.1	1.55	18372
AdamW [140]	94.4	0.425	16960
DP-AdamW [140]	92.3	2.33	21494
LoRA [71]	93.3	0.821	10366
DP-LoRA [71]	90.2	1.05	10496
LoRA [140]	94.3	0.301	11512
DP-LoRA [140]	91.3	0.332	11522
MeZO	92.5	0.345	2668
DPZero	91.8	0.347	2668

Comparison with DPGD-0th. In the previous synthetic example, DPGD-0th suffers from worse performance in larger dimensions. To provide a more complete comparison, we also evaluate the performance of DPGD-0th (Algorithm 1) for fine-tuning RoBERTa-large on the dataset TREC with a privacy budget of $\varepsilon = 2$ (the same setting as Table 2). DPGD-0th only achieves a test accuracy of 67.0, while DPZero attains 82.0. Moreover, DPGD-0th still requires per-sample clipping of the gradient estimator, which is costly in both memory and runtime compared to DPZero.

Clipping threshold. Our findings indicate that the optimal clipping threshold for DPZero tends to be higher than that for first-order methods. This observation aligns with the theoretical outcomes presented in Theorem 3, where the clipping threshold for DPZero is $C = \mathcal{O}(L\sqrt{\log(nd)})$, in contrast to the $\mathcal{O}(L)$ threshold adequate for first-order methods. In the concurrent study by [117], the chosen clipping threshold is 0.05. However, their implementation applies the clipping to the term $f(x + \lambda u; \xi) - f(x - \lambda u; \xi)$. After normalization by $\lambda = 10^{-3}$, it aligns with the order of magnitude used in our method. The validity of opting for a larger clipping threshold in DPZero is further confirmed through the private fine-tuning of RoBERTa (125M) on the SNLI dataset in Figure 3. An additional observation from our experiments is that the non-private baseline MeZO also appears to benefit from clipping. For instance, without clipping, the original MeZO encounters



(a) Training Loss. (b) Test Loss. (c) Test Accuracy (%).

Figure 3: Experiments on private fine-tuning RoBERTa (125M) for SNLI with DPZero. (a) (Smoothed) training curves when fixing the stepsize to be 5×10^{-6} and varying the clipping threshold from 1 to 500. In the choice of clipping, a tradeoff emerges; larger clipping values result in unnecessarily high privacy noise, while smaller values can induce increased bias in the optimization process. (b) and (c) Test loss and accuracy (%) when varying the stepsize and clipping threshold together. Consistent with first-order methods [71], we observe that larger clipping necessitates smaller stepsizes, whereas smaller clipping favors larger stepsizes.

non-convergence issues at a stepsize of 5×10^{-6}. Conversely, incorporating clipping permits the use of larger stepsizes and yields better results. A thorough investigation of this phenomenon is reserved for future research.

B.3 Private Fine-Tuning of the Language Model OPT

Table 10: Hyper-parameters used for fine-tuning OPT. We randomly sample 1000 samples for training and 1000 samples for testing. Stepsize and clipping are optimized through a grid search over the listed values. Other parameters are fixed.

Hyper-parameters	Values
Number of training samples	1000
Number of test samples	1000
Number of iterations	20000
Batch size	8
Privacy	$(\varepsilon = \{2, 6\}, \delta = 10^{-5})$
Smoothing λ	10^{-3}
Stepsize	$\{10^{-6}, 10^{-7}\}$
Clipping	{10, 50, 100, 200}

Table 11: Memory consumption (MiB) when fine-tuning OPT for BoolQ with batch size 8. All experiments are tested on a single GPU with 24 GiB memory. '-' in the table denotes out of memory. MeZO and DPZero can fit models up to OPT-6.7B, while the first-order method AdamW already runs out of memory on OPT-1.3B.

Method	OPT-1.3B	OPT-2.7B	OPT-6.7B	OPT-13B
AdamW	-	-	-	-
MeZO	7866	11602	20548	-
DPZero	7866	11602	20548	-

We follow experiment settings in Malladi et al. [87] to evaluate the performance of DPZero in the private fine-tuning of OPT [148] across four different datasets: SST-2 [114] for sentiment classification and BoolQ [25], SQuAD [103], and DROP [30] for question answering. In our experiments, we employ the same prompts as used in Malladi et al. [87] and use the same implementation as explained before. All results are averaged over

three random seeds $\{0, 29, 83\}$. The hyper-parameters used for our experiments are summarized in Table 10, and the memory usages on the dataset BoolQ are reported in Table 11.

C Technical Lemmas

Lemma C.1. *Let u be uniformly sampled from the Euclidean sphere $\sqrt{d}\,\mathbb{S}^{d-1}$, $a \in \mathbb{R}^d$ be some fixed vector independent of u, and $H \in \mathbb{R}^{d\times d}$ be some fixed matrix independent of u. We have that*

(i) $\mathbb{E}[u] = 0$ *and* $\mathbb{E}[uu^\top] = \mathrm{I}_d$.

(ii) $\mathbb{E}_u[u^\top a] = 0$, $\mathbb{E}_u[(u^\top a)^2] = \|a\|^2$ *and* $\forall\, C \geq 0$,

$$\mathbb{P}(|u^\top a| \geq C) \leq 2\sqrt{2\pi}\exp\left(-\frac{C^2}{8\|a\|^2}\right).$$

(iii) $\mathbb{E}_u[(u^\top a)u] = a$ *and*

$$\mathbb{E}_u[(u^\top a)^2\|u\|^2] = d\|a\|^2,$$
$$\mathbb{E}_u[(u^\top a)^2 uu^\top] = \frac{d}{d+2}\left(2aa^\top + \|a\|^2\mathrm{I}_d\right).$$

(iv) $\mathbb{E}_u[u^\top Hu] = \mathrm{Tr}(H)$ *and*

$$\mathbb{E}_u[(u^\top a)^2 u^\top Hu] = \frac{d}{d+2}\left(2a^\top Ha + \|a\|^2\,\mathrm{Tr}(H)\right).$$

Proof. (i) is a standard result, e.g., in Duchi et al. [32], and follows by the symmetry of the sphere. For any $u \in \sqrt{d}\cdot\mathbb{S}^{d-1}$, it must be the case that $-u \in \sqrt{d}\cdot\mathbb{S}^{d-1}$ as well, which suggests that $\mathbb{E}[u] = 0$. Since $\mathbb{E}[\sum_{i=1}^d u_i^2] = \mathbb{E}\|u\|^2 = d$, we immediately have that $\mathbb{E}[u_i^2] = 1$ for every i by symmetry. Then for the off-diagonal terms, since for any $u = (u_1, \cdots, u_i, \cdots, u_j, \cdots, u_d) \in \sqrt{d}\cdot\mathbb{S}^{d-1}$, it must be the case that $(u_1, \cdots, u_i, \cdots, -u_j, \cdots, u_d) \in \sqrt{d}\cdot\mathbb{S}^{d-1}$ as well, which suggests that $\mathbb{E}[u_iu_j] = 0$ when $i \neq j$. As a result, we can conclude that the matrix $\mathbb{E}[uu^\top] = \mathrm{I}_d$.

We then show (ii). Applying (i), we have that $\mathbb{E}_u[u^\top a] = 0$, and that

$$\begin{aligned}\mathbb{E}_u[(u^\top a)^2] &= \sum_{i=1}^d a_i^2\mathbb{E}[u_i^2] + \sum_{i\neq j} a_ia_j\mathbb{E}[u_iu_j]\\ &= \|a\|^2.\end{aligned}$$

The tail bound follows from Example 3.12 in Wainwright [128], where they showed that for any function $h : \mathbb{S}^{d-1} \to \mathbb{R}$ such that $\forall x, y \in \mathbb{S}^{d-1}$,

$$|h(x) - h(y)| \leq \arccos(x^\top y),$$

when x is uniformly sampled from $\mathbb{S}^{d-1}$, it holds that $\forall\,\gamma \geq 0$,

$$\mathbb{P}(|h(x) - \mathbb{E}[h(x)]| \geq \gamma) \leq 2\sqrt{2\pi}\exp\left(-\frac{d\gamma^2}{8}\right). \tag{8}$$

Let $h(x) = x^\top a/\|a\|$ for $x \in \mathbb{S}^{d-1}$. First, we have that $\forall x, y \in \mathbb{S}^{d-1}$,

$$\begin{aligned}|h(x) - h(y)|^2 &= \frac{|(x-y)^\top a|^2}{\|a\|^2}\\ &\leq \|x-y\|^2\\ &= 2(1 - x^\top y)\\ &\leq (\arccos(x^\top y))^2,\end{aligned}$$

where we use the inequality that $\theta^2/2 + \cos(\theta) - 1 \geq 0$ for $\theta \in [0, \pi]$ and let $x^\top y = \cos(\theta)$ such that $\arccos(x^\top y) = \theta$ for some $\theta \in [0, \pi]$. When u is uniformly sampled from $\sqrt{d} \cdot \mathbb{S}^{d-1}$, we know $u/\sqrt{d}$ is uniformly from $\mathbb{S}^{d-1}$. Applying (8) for $h(x) = x^\top a/\|a\|$ where $x \in \mathbb{S}^{d-1}$, we obtain that

$$\mathbb{P}\left(\left|\frac{u^\top a}{\sqrt{d}\|a\|} - \frac{\mathbb{E}[u^\top a]}{\sqrt{d}\|a\|}\right| \geq \gamma\right) \leq 2\sqrt{2\pi}\exp\left(-\frac{d\gamma^2}{8}\right).$$

Setting $C = \gamma\sqrt{d}\|a\|$, the proof is complete since $\mathbb{E}[u^\top a] = 0$. Similar results also exist in Theorem 5.1.4 of Vershynin [125], with all constants hidden behind some absolute c.

Next, we prove (iii). Applying (i), we have that

$$\begin{aligned}\mathbb{E}_u[(u^\top a)u_i] &= a_i\mathbb{E}[u_i^2] + \sum_{j\neq i} a_j\mathbb{E}[u_iu_j] \\ &= a_i.\end{aligned}$$

This implies that $\mathbb{E}_u[(u^\top a)u] = a$. Applying (ii), we obtain that

$$\begin{aligned}\mathbb{E}_u[(u^\top a)^2\|u\|^2] &= d \cdot \mathbb{E}_u[(u^\top a)^2] \\ &= d\|a\|^2.\end{aligned}$$

For the expectation of the matrix, we start from the diagonal terms.

$$\begin{aligned}\mathbb{E}_u[(u^\top a)^2u_i^2] &= \sum_{j=1}^d a_j^2\mathbb{E}[u_j^2u_i^2] + \sum_{j\neq k} a_ja_k\mathbb{E}[u_ju_ku_i^2] \\ &= a_i^2\mathbb{E}[u_i^4] + \sum_{j\neq i} a_j^2\mathbb{E}[u_j^2u_i^2].\end{aligned} \tag{9}$$

Here, we use the property that $\mathbb{E}[u_ju_ku_i^2] = 0$ for every i when $j \neq k$. This follows from symmetry of the sphere such that for any $u = (u_1, \cdots, u_j, \cdots, u_k, \cdots, u_d) \in \sqrt{d} \cdot \mathbb{S}^{d-1}$, it must be the case that $(u_1, \cdots, u_j, \cdots, -u_k, \cdots, u_d) \in \sqrt{d} \cdot \mathbb{S}^{d-1}$ as well. Again by symmetry, we have $\mathbb{E}[u_i^4]$ remains the same for every i, and $\mathbb{E}[u_i^2u_j^2]$ is the same for every $i \neq j$. Denote $w_1 = \mathbb{E}[u_i^4]$ and $w_2 = \mathbb{E}[u_i^2u_j^2]$. Since it holds that

$$\begin{aligned}\sum_{i=1}^d \mathbb{E}_u[(u^\top a)^2u_i^2] &= \mathbb{E}_u[(u^\top a)^2\|u\|^2] \\ &= d\|a\|^2,\end{aligned}$$

taking summation over (9), we can have that

$$\begin{aligned}d\|a\|^2 &= \sum_{i=1}^d a_i^2\mathbb{E}[u_i^4] + \sum_{i=1}^d\sum_{j=1,j\neq i}^d a_j^2\mathbb{E}[u_j^2u_i^2] \\ &= w_1\|a\|^2 + w_2\sum_{i=1}^d(\|a\|^2 - a_i^2) \\ &= w_1\|a\|^2 + (d-1)w_2\|a\|^2.\end{aligned}$$

This holds for arbitrary $a \in \mathbb{R}^d$, and thus we obtain that

$$w_1 + (d-1)w_2 = d. \tag{10}$$

We only compute $w_1 = \mathbb{E}[u_i^4]$ by showing that u_i^2/d actually follows the Beta distribution, and the value of w_2 can be derived from (10). First, $z/\|z\|$ is uniformly distributed on the unit sphere $\mathbb{S}^{d-1}$ for $z \in \mathbb{R}^d$ sampled from the standard multivariate Gaussian $\mathcal{N}(0, \mathrm{I}_d)$ [92, 89]. This means that z_i^2 is distributed according to the χ^2-distribution with 1 degree of freedom, and $\bar{z}_i^2 := \sum_{j\neq i} z_j^2$ is distributed according to the χ^2-distribution with

degree $(d-1)$. Since χ^2-distribution is a special case of the Gamma distribution and z_i^2, $\bar{z}_i^2$ are independent, we conclude that $z_i^2/(z_i^2+\bar{z}_i^2)$ has the Beta distribution with parameters $1/2$ and $(d-1)/2$ [26, 50]. Finally, since $u/\sqrt{d}$ is uniformly distributed on $\mathbb{S}^{d-1}$, by symmetry of the sphere, we know that u_i^2/d has the same Beta distribution as $z_i^2/(z_i^2+\bar{z}_i^2)$. The mean and variance of $\mathrm{Beta}(1/2,(d-1)/2)$ is $1/d$ and $2(d-1)/(d^2(d+2))$. This suggests that $\mathbb{E}[u_i^2]=1$, as already proved in (i), and that

$$\begin{aligned} w_1 &= \mathbb{E}[(u_i^2-\mathbb{E}[u_i^2])^2]+(\mathbb{E}[u_i^2])^2 \\ &= d^2\left(\frac{2(d-1)}{d^2(d+2)}+\frac{1}{d^2}\right) \\ &= \frac{3d}{d+2}. \end{aligned}$$

By (10), we know $w_2 = d/(d+2)$. According to (9), we have that the diagonal terms

$$\begin{aligned} \mathbb{E}_u[(u^\top a)^2 u_i^2] &= w_1 a_i^2 + w_2(\|a\|^2 - a_i^2) \\ &= \frac{2d}{d+2}a_i^2 + \frac{d}{d+2}\|a\|^2. \end{aligned}$$

Then we compute the off-diagonal entries for $i \neq j$. By the same reasoning as (9), we have that

$$\begin{aligned} \mathbb{E}_u[(u^\top a)^2 u_i u_j] &= \sum_{i\neq j} a_i a_j \mathbb{E}[u_i^2 u_j^2] \\ &= \frac{2d}{d+2}a_i a_j. \end{aligned}$$

All other terms equal to 0 by symmetry of the sphere. Combining both diagonal and off-diagonal elements, we have that $\mathbb{E}_u[(u^\top a)^2 uu^\top] = (d/(d+2))(2aa^\top + \|a\|^2 \mathrm{I}_d)$. Similar results are also shown in Appendix F of Malladi et al. [87].

Finally, we give the proof of (iv). For the first statement, applying (i) in this lemma, we have that

$$\begin{aligned} \mathbb{E}_u\left[u^\top H u\right] &= \mathbb{E}\Big[\operatorname{Tr}(uu^\top H)\Big] \\ &= \operatorname{Tr}\Big(\mathbb{E}[uu^\top]\cdot H\Big) \\ &= \operatorname{Tr}(H). \end{aligned}$$

Similarly for the second statement, we apply (iii) in this lemma and obtain that

$$\begin{aligned} \mathbb{E}_u\left[(u^\top a)^2 u^\top H u\right] &= \mathbb{E}\Big[(u^\top a)^2 \cdot \operatorname{Tr}(uu^\top H)\Big] \\ &= \mathbb{E}\Big[\operatorname{Tr}\Big((u^\top a)^2 uu^\top \cdot H\Big)\Big] \\ &= \operatorname{Tr}\Big(\mathbb{E}\Big[(u^\top a)^2 uu^\top\Big]\cdot H\Big) \\ &= \frac{2d}{d+2}\operatorname{Tr}(aa^\top H) + \frac{d}{d+2}\|a\|^2\operatorname{Tr}(H) \\ &= \frac{2d}{d+2}a^\top H a + \frac{d}{d+2}\|a\|^2\operatorname{Tr}(H). \end{aligned}$$

This concludes the proof. □

Lemma C.2. *Let u be uniformly sampled from the Euclidean sphere $\sqrt{d}\,\mathbb{S}^{d-1}$ and v be uniformly sampled from the Euclidean ball $\sqrt{d}\,\mathbb{B}^d = \{x\in\mathbb{R}^d \mid \|x\| \le \sqrt{d}\}$. For any function $f(x):\mathbb{R}^d\to\mathbb{R}$ and $\lambda>0$, we define its zeroth-order gradient estimator as $g_\lambda(x) = ((f(x+\lambda u)-f(x-\lambda u))/(2\lambda))u$ and the smoothed function as $f_\lambda(x) = \mathbb{E}_v[f(x+\lambda v)]$. The following properties hold:*

(i) $f_\lambda(x)$ is differentiable and $\mathbb{E}_u[g_\lambda(x)] = \nabla f_\lambda(x)$.

(ii) If $f(x)$ is ℓ-smooth, then we have that

$$\begin{aligned}\|\nabla f(x) - \nabla f_\lambda(x)\| &\leq \frac{\ell}{2}\lambda d^{3/2},\\ \mathbb{E}_u[\,\|g_\lambda(x)\|^2\,] &\leq 2d\cdot\|\nabla f(x)\|^2 + \frac{\ell^2}{2}\lambda^2 d^3.\end{aligned}$$

The above results are consistent with (iii) in Lemma C.1 when $\lambda \to 0$ and $f(x)$ is differentiable such that the two-point estimator reduces to the directional derivative $g_0(x) = u^\top \nabla f(x) u$.

Proof. We first show (i). Similarly to Lemma 10 in Shamir [108], we have that

$$\mathbb{E}_{u\in\sqrt{d}\cdot\mathbb{S}^{d-1}}[g_\lambda(x)] = \mathbb{E}_{u\in\sqrt{d}\cdot\mathbb{S}^{d-1}}\left[\frac{f(x+\lambda u)u}{\lambda}\right].$$

Applying Lemma 2.1 in Flaxman et al. [39], we know

$$\mathbb{E}_{u'\in\mathbb{S}^{d-1}}[f(x+\lambda' u')u'] = \frac{\lambda'}{d}\nabla\mathbb{E}_{v'\in\mathbb{B}^d}[f(x+\lambda' v')].$$

Introducing $u = \sqrt{d}u'$, $v = \sqrt{d}v'$ and $\lambda = \lambda'/\sqrt{d}$, we thus obtain

$$\begin{aligned}\mathbb{E}_{u\in\sqrt{d}\cdot\mathbb{S}^{d-1}}\left[\frac{f(x+\lambda u)u}{\lambda}\right] &= \mathbb{E}_{u'\in\mathbb{S}^{d-1}}\left[\frac{f(x+\lambda' u')u'd}{\lambda'}\right]\\ &= \nabla\mathbb{E}_{v'\in\mathbb{B}^d}[f(x+\lambda' v')]\\ &= \nabla\mathbb{E}_{v\in\sqrt{d}\cdot\mathbb{B}^d}[f(x+\lambda v)].\end{aligned}$$

The proof of (ii) mostly follows from Nesterov and Spokoiny [94], where the results are originally obtained for the case that u is sampled from the standard multivariate Gaussian distribution. By (iii) in Lemma C.1 and (i) here, we have that for u uniformly sampled from $\sqrt{d}\cdot\mathbb{S}^{d-1}$,

$$\begin{aligned}\|\nabla f(x) - \nabla f_\lambda(x)\| &= \left\|\mathbb{E}_u[(u^\top\nabla f(x))u] - \mathbb{E}_u\left[\frac{f(x+\lambda u)-f(x-\lambda u)}{2\lambda}u\right]\right\|\\ &\leq \mathbb{E}_u\left\|\left(\frac{f(x+\lambda u)-f(x-\lambda u)}{2\lambda} - u^\top\nabla f(x)\right)u\right\|\\ &\leq \frac{\sqrt{d}}{2\lambda}\mathbb{E}_u|f(x+\lambda u) - f(x) - \lambda u^\top\nabla f(x)|\\ &\quad + \frac{\sqrt{d}}{2\lambda}\mathbb{E}_u|f(x) - f(x-\lambda u) - \lambda u^\top\nabla f(x)|\\ &\leq \frac{\ell}{2}\lambda d^{3/2},\end{aligned}$$

where in the last step we use smoothness of $f(x)$ such that $|f(x+\lambda u) - f(x) - \lambda u^\top\nabla f(x)| \leq \ell\lambda^2 d/2$ and the same holds for $|f(x) - f(x-\lambda u) - \lambda u^\top\nabla f(x)| = |f(x-\lambda u) - f(x) + \lambda u^\top\nabla f(x)|$. The last statement holds similarly:

$$\begin{aligned}\mathbb{E}_u[\|g_\lambda(x)\|^2] &= \frac{d}{4\lambda^2}\mathbb{E}_u[(f(x+\lambda u) - f(x-\lambda u))^2]\\ &\leq 2d\cdot\mathbb{E}_u[(u^\top\nabla f(x))^2] + \frac{d}{2\lambda^2}\mathbb{E}_u[(f(x+\lambda u) - f(x-\lambda u) - 2\lambda u^\top\nabla f(x))^2]\\ &\leq 2d\cdot\mathbb{E}_u[(u^\top\nabla f(x))^2] + \frac{d}{\lambda^2}\mathbb{E}_u[(f(x+\lambda u) - f(x) - \lambda u^\top\nabla f(x))^2]\\ &\quad + \frac{d}{\lambda^2}\mathbb{E}_u[(f(x) - f(x-\lambda u) - \lambda u^\top\nabla f(x))^2]\\ &\leq 2d\cdot\|\nabla f(x)\|^2 + \frac{\ell^2}{2}\lambda^2 d^3,\end{aligned} \tag{11}$$

where in the last step we use Lemma C.1 and smoothness of $f(x)$. □

D Detailed Proof and Analysis of DPGD-0th (Algorithm 1)

Proof of Theorem 1. The privacy guarantees directly follow from Lemma 2.2 noticing that the sensitivity is $2C/n$. Note that the original advanced composition theorem in Kairouz et al. [62] is stated for the case where the output of $\mathcal{A}$ is a scalar. Given the spherical symmetry properties of Gaussian noise, the results can be readily extended to multiple dimensions, as outlined in Lemma 1 of Kenthapadi et al. [65] where the basis can be selected in a way such that $\mathcal{A}(S)$ and $\mathcal{A}(S')$ differ in exactly one dimension.

We then focus on the utility guarantee on $\mathbb{E}[\|\nabla F_S(x_\tau)\|^2]$. Since $f(x;\xi)$ is L-Lipschitz for every ξ by Assumption 3.1 and $\|u_t\| = \sqrt{d}$ by its construction, we have that

$$\begin{aligned}\|g_\lambda(x_t;\xi_i)\| &= \frac{|f(x_t+\lambda u_t;\xi_i) - f(x_t-\lambda u_t;\xi_i)|}{2\lambda}\|u_t\| \\ &\le L\|u_t\|^2 \\ &= Ld.\end{aligned}$$

This means $\mathrm{clip}_C(g_\lambda(x_t;\xi_i)) = g_\lambda(x_t;\xi_i)$ when setting $C = Ld$. For notation simplicity, we let

$$\begin{aligned}G_\lambda(x_t) &:= \frac{1}{n}\sum_{i=1}^n g_\lambda(x_t;\xi_i) \\ &= \frac{1}{n}\sum_{i=1}^n \frac{f(x_t+\lambda u_t;\xi_i) - f(x_t-\lambda u_t;\xi_i)}{2\lambda}u_t \\ &= \frac{F_S(x_t+\lambda u_t) - F_S(x_t-\lambda u_t)}{2\lambda}u_t.\end{aligned}$$

Algorithm 1 reduces to $x_{t+1} = x_t - \alpha(G_\lambda(x_t) + z_t)$. By smoothness of $F_S(x)$, we have that

$$\begin{aligned}F_S(x_{t+1}) &\le F_S(x_t) + \nabla F_S(x_t)^\top(x_{t+1}-x_t) + \frac{\ell}{2}\|x_{t+1}-x_t\|^2 \\ &= F_S(x_t) - \alpha\nabla F_S(x_t)^\top(G_\lambda(x_t)+z_t) + \frac{\ell}{2}\alpha^2\|G_\lambda(x_t)\|^2 + \frac{\ell}{2}\alpha^2\|z_t\|^2 + \ell\alpha^2 z_t^\top G_\lambda(x_t).\end{aligned}$$

Since z_t is sampled from $\mathcal{N}(0,\sigma^2\mathrm{I}_d)$ and is independent of x_t, u_t and S, we have that

$$\mathbb{E}_{z_t}[F_S(x_{t+1})] \le F_S(x_t) - \alpha\nabla F_S(x_t)^\top G_\lambda(x_t) + \frac{\ell}{2}\alpha^2\|G_\lambda(x_t)\|^2 + \frac{\ell}{2}\alpha^2\, d\,\sigma^2.$$

Define $F_\lambda(x) := \mathbb{E}_v[F_S(x+\lambda v)]$ for v sampled uniformly from the Euclidean ball $\sqrt{d}\cdot\mathbb{B}^d$. By Lemma C.2, we know $\mathbb{E}_{u_t}[G_\lambda(x_t)] = \nabla F_\lambda(x_t)$. Since u_t is independent of x_t and S, taking expectation with respect to u_t and applying (ii) in Lemma C.2, we obtain that

$$\begin{aligned}\mathbb{E}_{z_t,u_t}[F_S(x_{t+1})] &\le F_S(x_t) - \alpha\nabla F_S(x_t)^\top\nabla F_\lambda(x_t) + \frac{\ell}{2}\alpha^2\mathbb{E}_{u_t}[\|G_\lambda(x_t)\|^2] + \frac{\ell}{2}\alpha^2\, d\,\sigma^2 \\ &= F_S(x_t) - \frac{\alpha}{2}\|\nabla F_S(x_t)\|^2 - \frac{\alpha}{2}\|\nabla F_\lambda(x_t)\|^2 + \frac{\alpha}{2}\|\nabla F_\lambda(x_t) - \nabla F_S(x_t)\|^2 \\ &\quad + \frac{\ell}{2}\alpha^2\,\mathbb{E}_{u_t}[\|G_\lambda(x_t)\|^2] + \frac{\ell}{2}\alpha^2\, d\,\sigma^2 \\ &\le F_S(x_t) - \frac{\alpha}{2}(1-2d\ell\alpha)\|\nabla F_S(x_t)\|^2 + \frac{\ell^2}{8}\alpha(1+2\ell\alpha)\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, d\,\sigma^2. \qquad (12)\end{aligned}$$

Choosing $\alpha = 1/(4\ell d)$ such that $1 - 2d\ell\alpha = 1/2$ and $2\ell\alpha < 1$, we obtain that

$$\begin{aligned}\mathbb{E}[\|\nabla F_S(x_t)\|^2] &< \frac{4\,\mathbb{E}[F_S(x_t) - F_S(x_{t+1})]}{\alpha} + \ell^2\lambda^2 d^3 + 2\ell\alpha\, d\sigma^2 \\ &= \frac{4\,\mathbb{E}[F_S(x_t) - F_S(x_{t+1})]}{\alpha} + \ell^2\lambda^2 d^3 + \frac{64\ell C^2\,\alpha T\, d\log(e + (\varepsilon/\delta))}{n^2\varepsilon^2} \\ &= \frac{4\,\mathbb{E}[F_S(x_t) - F_S(x_{t+1})]}{\alpha} + \ell^2\lambda^2 d^3 + \frac{64\ell L^2\,\alpha T\, d^3\log(e + (\varepsilon/\delta))}{n^2\varepsilon^2}.\end{aligned}$$

As a result, taking summation from $t=0$ to $T-1$ and dividing both sides by T, we have that

$$\begin{aligned}\mathbb{E}[\|\nabla F_S(x_\tau)\|^2] &= \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F_S(x_t)\|^2] \\ &\le \frac{4(F_S(x_0)-F_S^*)}{\alpha T} + \ell^2\lambda^2 d^3 + \frac{64\ell L^2\,\alpha T\, d^3\log(e+(\varepsilon/\delta))}{n^2\varepsilon^2} \\ &\le \frac{16(\ell(F_S(x_0)-F_S^*)+2L^2)d\sqrt{d\log(e+(\varepsilon/\delta))}}{n\varepsilon},\end{aligned}$$

with the choice of parameters

$$\alpha T = \frac{n\varepsilon}{4\ell d\sqrt{d\log(e+(\varepsilon/\delta))}}, \quad \lambda \le \frac{4L}{\ell d}\left(\frac{\sqrt{d\log(e+(\varepsilon/\delta))}}{n\varepsilon}\right)^{1/2}.$$

This suggests that the total number of iteration is $T = n\varepsilon/\sqrt{d\log(e+(\varepsilon/\delta))}$ and the total number of zeroth-order gradient computations is $nT = n^2\varepsilon/\sqrt{d\log(e+(\varepsilon/\delta))}$. Note that the above selection of parameters ensures scale invariance. □

Proof of Theorem 2. The privacy analysis remains the same as before, and we focus on the utility analysis on $\mathbb{E}\|\nabla F_S(x_\tau)\|^2$. By the same reasoning, when setting $C = Ld$, Algorithm 1 reduces to $x_{t+1} = x_t - \alpha(G_\lambda(x_t)+z_t)$ where $G_\lambda(x_t) = (F_S(x_t+\lambda u_t) - F_S(x_t-\lambda u_t))u_t/(2\lambda)$. By Taylor's theorem with remainder, for some $\theta\in(0,1)$, we have that

$$\begin{aligned}F_S(x_{t+1}) &= F_S(x_t) + \nabla F_S(x_t)^\top(x_{t+1}-x_t) + \frac{1}{2}(x_{t+1}-x_t)^\top\nabla^2 F_S(x_t+\theta(x_{t+1}-x_t))(x_{t+1}-x_t) \\ &\le F_S(x_t) - \alpha\nabla F_S(x_t)^\top\left(G_\lambda(x_t)+z_t\right) + \frac{\alpha^2}{2}G_\lambda(x_t)^\top H G_\lambda(x_t) + \frac{\alpha^2}{2}z_t^\top H z_t \\ &\quad + \frac{\alpha^2}{2}\left(G_\lambda(x_t)^\top H z_t + z_t^\top H G_\lambda(x_t)\right).\end{aligned}$$

Here in the inequality, we use Assumption 3.5 such that $\nabla^2 F_S(x) \preceq H$ for any $x\in\mathbb{R}^d$. Similarly to (iv) in Lemma C.1, we have that $\mathbb{E}[z_t^\top H z_t] = \operatorname{Tr}(\mathbb{E}[z_t z_t^\top]H) = \sigma^2\operatorname{Tr}(H)$. Since z_t is sampled from $\mathcal{N}(0,\sigma^2\mathrm{I}_d)$ and is independent of u_t, x_t and the dataset S, taking expectation with respect to z_t, we can then obtain that

$$\begin{aligned}\mathbb{E}_{z_t}[F_S(x_{t+1})] &\le F_S(x_t) - \alpha\nabla F_S(x_t)^\top G_\lambda(x_t) + \frac{\alpha^2}{2}G_\lambda(x_t)^\top H G_\lambda(x_t) + \frac{\alpha^2}{2}\mathbb{E}_{z_t}[z_t^\top H z_t] \\ &= F_S(x_t) - \alpha\nabla F_S(x_t)^\top G_\lambda(x_t) + \frac{\alpha^2}{2}G_\lambda(x_t)^\top H G_\lambda(x_t) + \frac{\alpha^2\sigma^2}{2}\operatorname{Tr}(H).\end{aligned} \tag{13}$$

Assumption 3.5 implies $F_S(x)$ is also ℓ-smooth. By a similar argument as (11) in the proof of (ii) in Lemma C.2, we have

$$\left(\frac{F_S(x_t+\lambda u_t)-F_S(x_t-\lambda u_t)}{2\lambda}\right)^2 \le 2\left(u_t^\top\nabla F_S(x_t)\right)^2 + \frac{\ell^2}{2}\lambda^2 d^2. \tag{14}$$

As $u_t^\top H u_t \ge 0$, by (iv) in Lemma C.1 and Assumption 3.5, we have that

$$\begin{aligned}\mathbb{E}\left[G_\lambda(x_t)^\top H G_\lambda(x_t)\right] &= \mathbb{E}\left[\left(\frac{F_S(x_t+\lambda u_t)-F_S(x_t-\lambda u_t)}{2\lambda}\right)^2 u_t^\top H u_t\right] \\ &\le 2\,\mathbb{E}\left[\left(u_t^\top\nabla F_S(x_t)\right)^2 u_t^\top H u_t\right] + \frac{\ell^2}{2}\lambda^2 d^2\,\mathbb{E}\left[u_t^\top H u_t\right] \\ &= \frac{2d}{d+2}\left(2\nabla F_S(x_t)^\top H\nabla F_S(x_t) + \|\nabla F_S(x_t)\|^2\operatorname{Tr}(H)\right) + \frac{\ell^2}{2}\lambda^2 d^2\operatorname{Tr}(H) \\ &\le 2\ell(r+2)\|\nabla F_S(x_t)\|^2 + \frac{\ell^3}{2}\lambda^2 d^2 r.\end{aligned}$$

Taking expectation of (13) with respect to u_t, by Lemma C.2 for $F_\lambda(x) = \mathbb{E}_v[F_S(x+\lambda v)]$ with v uniformly sampled from $\sqrt{d}\cdot\mathbb{B}^d$, we have that

$$\begin{aligned}\mathbb{E}[F_S(x_{t+1})] &\le F_S(x_t) - \alpha\nabla F_S(x_t)^\top \nabla F_\lambda(x_t) + \ell\alpha^2(r+2)\|\nabla F_S(x_t)\|^2 + \frac{\ell^3\alpha^2\lambda^2 d^2 r}{4} + \frac{\ell\alpha^2 r\sigma^2}{2}\\ &\le F_S(x_t) - \frac{\alpha}{2}(1-2(r+2)\ell\alpha)\|\nabla F_S(x_t)\|^2 + \frac{\alpha}{2}\|\nabla F_S(x_t) - \nabla F_\lambda(x_t)\|^2 + \frac{\ell^3\alpha^2\lambda^2 d^2 r}{4} + \frac{\ell\alpha^2 r\sigma^2}{2}\\ &\le F_S(x_t) - \frac{\alpha}{2}(1-2(r+2)\ell\alpha)\|\nabla F_S(x_t)\|^2 + \frac{\ell^2\alpha\lambda^2 d^2(d+2r\ell\alpha)}{8} + \frac{\ell\alpha^2 r\sigma^2}{2}. \end{aligned} \tag{15}$$

Choosing $\alpha = 1/(4\ell(r+2))$ such that $1-2(r+2)\ell\alpha = 1/2$ and $2\ell\alpha r < 1 \le d$, we have that

$$\begin{aligned}\mathbb{E}[\|\nabla F_S(x_t)\|^2] &< \frac{4\,\mathbb{E}[F_S(x_t) - F_S(x_{t+1})]}{\alpha} + \ell^2\lambda^2 d^3 + 2\ell\alpha\, r\sigma^2\\ &= \frac{4\,\mathbb{E}[F_S(x_t) - F_S(x_{t+1})]}{\alpha} + \ell^2\lambda^2 d^3 + \frac{64\ell C^2\,\alpha T\, r\log(e+(\varepsilon/\delta))}{n^2\varepsilon^2}\\ &= \frac{4\,\mathbb{E}[F_S(x_t) - F_S(x_{t+1})]}{\alpha} + \ell^2\lambda^2 d^3 + \frac{64\ell L^2\,\alpha T\, d^2 r\log(e+(\varepsilon/\delta))}{n^2\varepsilon^2}.\end{aligned}$$

As a result, taking summation from $t=0$ to $T-1$ and dividing both sides by T, we have that

$$\begin{aligned}\mathbb{E}[\|\nabla F_S(x_\tau)\|^2] &= \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F_S(x_t)\|^2]\\ &\le \frac{4(F_S(x_0) - F_S^*)}{\alpha T} + \ell^2\lambda^2 d^3 + \frac{64\ell L^2\,\alpha T\, d^2 r\log(e+(\varepsilon/\delta))}{n^2\varepsilon^2}\\ &\le \frac{16(\ell(F_S(x_0) - F_S^*) + 2L^2)d\sqrt{r\log(e+(\varepsilon/\delta))}}{n\varepsilon},\end{aligned}$$

with the choice of parameters

$$\alpha T = \frac{n\varepsilon}{4\ell d\sqrt{r\log(e+(\varepsilon/\delta))}}, \quad \lambda \le \frac{4L}{\ell d}\left(\frac{\sqrt{r\log(e+(\varepsilon/\delta))}}{n\varepsilon}\right)^{1/2}.$$

This suggests that the total number of iteration is $T = n(r+2)\varepsilon/(d\sqrt{r\log(e+(\varepsilon/\delta))})$ and the total number of zeroth-order gradient computations is $nT = n^2(r+2)\varepsilon/(d\sqrt{r\log(e+(\varepsilon/\delta))})$. The above selection ensures scale invariance. □

E Detailed Proof and Analysis of DPZero (Algorithm 2)

Privacy guarantee. Since u_t is independent of the dataset S, the privacy guarantees directly follow from Lemma 2.2 and post-processing [34] noticing that the sensitivity is $2C/n$. We want to emphasis that the randomness of u_t is never used for the privacy guarantee, and the analysis holds for any u_t as long as it is independent of the dataset.

Utility guarantee. We then focus on the utility guarantee on $\mathbb{E}\|\nabla F_S(x_\tau)\|^2$. Since $f(x;\xi)$ is ℓ-smooth for every ξ by Assumption 3.5, we have that

$$\begin{aligned}\frac{|f(x_t+\lambda u_t;\xi_i) - f(x_t-\lambda u_t;\xi_i)|}{2\lambda} &\le |u_t^\top\nabla f(x_t;\xi_i)| + \frac{|f(x_t+\lambda u_t;\xi_i) - f(x_t;\xi_i) - \lambda u_t^\top\nabla f(x_t;\xi_i)|}{2\lambda}\\ &\quad + \frac{|f(x_t-\lambda u_t;\xi_i) - f(x_t;\xi_i) + \lambda u_t^\top\nabla f(x_t;\xi_i)|}{2\lambda}\\ &\le |u_t^\top\nabla f(x_t;\xi_i)| + \frac{\ell}{2}\lambda d.\end{aligned} \tag{16}$$

Therefore, by (ii) in Lemma C.1 and Lipschitzness of $f(x;\xi)$, we have that

$$\begin{aligned}\mathbb{P}\left(\frac{|f(x_t+\lambda u_t;\xi_i)-f(x_t-\lambda u_t;\xi_i)|}{2\lambda}\geq C_0+\frac{\ell}{2}\lambda d\right)&\leq \mathbb{P}(|u_t^\top\nabla f(x_t;\xi_i)|\geq C_0)\\&\leq 2\sqrt{2\pi}\exp\left(-\frac{C_0^2}{8\|\nabla f(x_t;\xi_i)\|^2}\right)\\&\leq 2\sqrt{2\pi}\exp\left(-\frac{C_0^2}{8L^2}\right).\end{aligned}$$

We define $Q_{t,i}$ to be the event that the clipping does not happen at iteration t for sample ξ_i and $\bar{Q}_{t,i}$ to be the event that the clipping does happen. The above equation implies that if the clipping threshold $C\geq C_0+\ell\lambda d/2$, then we have that $\mathbb{P}(\bar{Q}_{t,i})\leq 2\sqrt{2\pi}\exp(-C_0^2/(8L^2))$. Let Q_t denote the event that the clipping does not happen at iteration t for every sample $1\leq i\leq n$, and let $\bar{Q}_t$ be the event that there exist some i such that the clipping does happen at iteration t. We also denote Q as the event that the clipping does not happen for every iteration $t=0,1,\cdots,T-1$ and every sample $1\leq i\leq n$ and $\bar{Q}$ as the event that there exist some t and i such that the clipping does happen. By the union bound, we have that

$$\begin{aligned}\mathbb{P}(\bar{Q})&=\mathbb{P}\left(\bigcup_{t=0}^{T-1}\bigcup_{i=1}^{n}\bar{Q}_{t,i}\right)\\&\leq 2\sqrt{2\pi}\,nT\exp\left(-\frac{C_0^2}{8L^2}\right).\end{aligned}$$

To simplify the notation, we let

$$\begin{aligned}G_\lambda(x_t)&=\frac{1}{n}\sum_{i=1}^{n}\frac{f(x_t+\lambda u_t;\xi_i)-f(x_t-\lambda u_t;\xi_i)}{2\lambda}u_t\\&=\frac{F_S(x_t+\lambda u_t)-F_S(x_t-\lambda u_t)}{2\lambda}u_t,\end{aligned}$$

and its per-sample clipped version as

$$\hat{G}_\lambda(x_t)=\frac{1}{n}\sum_{i=1}^{n}\mathrm{clip}_C\left(\frac{f(x_t+\lambda u_t;\xi_i)-f(x_t-\lambda u_t;\xi_i)}{2\lambda}\right)u_t.$$

Algorithm 2 becomes $x_{t+1}=x_t-\alpha(\hat{G}_\lambda(x_t)+z_tu_t)$ under the above notation. By Taylor's theorem with remainder, for some $\theta\in(0,1)$, we have that

$$\begin{aligned}F_S(x_{t+1})&=F_S(x_t)+\nabla F_S(x_t)^\top(x_{t+1}-x_t)+\frac{1}{2}(x_{t+1}-x_t)^\top\nabla^2F_S(x_t+\theta(x_{t+1}-x_t))(x_{t+1}-x_t)\\&\leq F_S(x_t)-\alpha\nabla F_S(x_t)^\top\left(\hat{G}_\lambda(x_t)+z_tu_t\right)+\frac{\alpha^2}{2}\hat{G}_\lambda(x_t)^\top H\hat{G}_\lambda(x_t)+\frac{\alpha^2}{2}z_t^2u_t^\top Hu_t\\&\quad+\frac{\alpha^2}{2}z_t\left(\hat{G}_\lambda(x_t)^\top Hu_t+u_t^\top H\hat{G}_\lambda(x_t)\right).\end{aligned}$$

Here in the inequality, we use Assumption 3.5 such that $\nabla^2F_S(x)\preceq H$ for any $x\in\mathbb{R}^d$. The event Q_t depends on the randomness in $u_{<(t+1)}:=\{u_0,u_1,\cdots,u_t\}$ and $z_{<t}:=\{z_0,z_1,\cdots,z_{t-1}\}$. Note that the scalar noise z_t sampled from $\mathcal{N}(0,\sigma^2)$ is independent of $u_{<(t+1)}$, $z_{<t}$, x_t, and the dataset S. Conditioned on the event Q_t and taking expectation with respect to $z_{<(t+1)}$ and $u_{<(t+1)}$, we have that

$$\begin{aligned}\mathbb{E}_{z_{<(t+1)},u_{<(t+1)}}[F_S(x_{t+1})|Q_t]&\leq\mathbb{E}_{z_{<t},u_{<t}}[F_S(x_t)|Q_t]-\alpha\,\mathbb{E}_{z_{<t},u_{<(t+1)}}\left[\nabla F_S(x_t)^\top\hat{G}_\lambda(x_t)\middle|Q_t\right]\\&\quad+\frac{\alpha^2}{2}\,\mathbb{E}_{z_{<t},u_{<(t+1)}}\left[\hat{G}_\lambda(x_t)^\top H\hat{G}_\lambda(x_t)\middle|Q_t\right]+\frac{\alpha^2\sigma^2}{2}\,\mathbb{E}_{z_{<t},u_{<(t+1)}}\left[u_t^\top Hu_t\middle|Q_t\right].\end{aligned}\tag{17}$$

Let $\mathbb{E}_t := \mathbb{E}_{z_{<t}, u_{<(t+1)}}$ for simplicity. Given the condition that Q_t happens, we know that $\hat{G}_\lambda(x_t) = G_\lambda(x_t)$ and

$$\mathbb{E}_t\left[\hat{G}_\lambda(x_t)^\top H \hat{G}_\lambda(x_t) \,\middle|\, Q_t\right] = \mathbb{E}_t\left[\left(\frac{F_S(x_t+\lambda u_t) - F_S(x_t-\lambda u_t)}{2\lambda}\right)^2 u_t^\top H u_t \,\middle|\, Q_t\right].$$

Since $H \succeq 0$, we have that $u_t^\top H u_t \geq 0$. By the law of total probability, we obtain

$$\begin{aligned}
&\mathbb{E}_t\left[\left(\frac{F_S(x_t+\lambda u_t) - F_S(x_t-\lambda u_t)}{2\lambda}\right)^2 u_t^\top H u_t\right] \\
&= \mathbb{E}_t\left[\left(\frac{F_S(x_t+\lambda u_t) - F_S(x_t-\lambda u_t)}{2\lambda}\right)^2 u_t^\top H u_t \,\middle|\, Q_t\right]\mathbb{P}(Q_t) \\
&\quad + \mathbb{E}_t\left[\left(\frac{F_S(x_t+\lambda u_t) - F_S(x_t-\lambda u_t)}{2\lambda}\right)^2 u_t^\top H u_t \,\middle|\, \bar{Q}_t\right]\mathbb{P}(\bar{Q}_t) \\
&\geq \mathbb{E}_t\left[\left(\frac{F_S(x_t+\lambda u_t) - F_S(x_t-\lambda u_t)}{2\lambda}\right)^2 u_t^\top H u_t \,\middle|\, Q_t\right]\mathbb{P}(Q_t).
\end{aligned} \tag{18}$$

Assumption 3.5 implies $F_S(x)$ is also ℓ-smooth. Similarly to the proof of Theorem 2, by (14) and the fact that $u_t^\top H u_t \geq 0$, applying (iv) in Lemma C.1 and Assumption 3.5, we can then obtain that

$$\begin{aligned}
\mathbb{E}_t\left[\hat{G}_\lambda(x_t)^\top H \hat{G}_\lambda(x_t) \,\middle|\, Q_t\right] &\leq \frac{\mathbb{E}_t\left[(F_S(x_t+\lambda u_t) - F_S(x_t-\lambda u_t))^2 u_t^\top H u_t\right]}{4\lambda^2 \cdot \mathbb{P}(Q_t)} \\
&\leq \frac{\mathbb{E}_t\left[2\left(u_t^\top \nabla F_S(x_t)\right)^2 u_t^\top H u_t\right]}{\mathbb{P}(Q_t)} + \frac{\ell^2\lambda^2 d^2}{2\,\mathbb{P}(Q_t)}\mathbb{E}_t\left[u_t^\top H u_t\right] \\
&= \frac{2d\,\mathbb{E}_{z_{<t}, u_{<t}}\left[2\nabla F_S(x_t)^\top H \nabla F_S(x_t) + \|\nabla F_S(x_t)\|^2 \operatorname{Tr}(H)\right]}{(d+2)\,\mathbb{P}(Q_t)} + \frac{\ell^2\lambda^2 d^2 \operatorname{Tr}(H)}{2\,\mathbb{P}(Q_t)} \\
&\leq \frac{2\ell(r+2)}{\mathbb{P}(Q_t)}\mathbb{E}_{z_{<t}, u_{<t}}\|\nabla F_S(x_t)\|^2 + \frac{\ell^3\lambda^2 d^2 r}{2\,\mathbb{P}(Q_t)}.
\end{aligned} \tag{19}$$

The same as (18), we can also get that

$$\begin{aligned}
\mathbb{E}_t\left[u_t^\top H u_t \,\middle|\, Q_t\right] &\leq \frac{\mathbb{E}_t\left[u_t^\top H u_t\right]}{\mathbb{P}(Q_t)} \\
&\leq \frac{r\ell}{\mathbb{P}(Q_t)}.
\end{aligned} \tag{20}$$

For the inner-product term, we have that

$$\mathbb{E}_t\left[\nabla F_S(x_t)^\top \hat{G}_\lambda(x_t) \,\middle|\, Q_t\right] = \mathbb{E}_t\left[\nabla F_S(x_t)^\top G_\lambda(x_t) \,\middle|\, Q_t\right].$$

By the law of total probability, since u_t is independent of x_t, we know that

$$\begin{aligned}
\mathbb{E}_t\left[\nabla F_S(x_t)^\top G_\lambda(x_t) \,\middle|\, Q_t\right]\mathbb{P}(Q_t) + \mathbb{E}_t\left[\nabla F_S(x_t)^\top G_\lambda(x_t) \,\middle|\, \bar{Q}_t\right]\mathbb{P}(\bar{Q}_t) &= \mathbb{E}_t\left[\nabla F_S(x_t)^\top G_\lambda(x_t)\right] \\
&= \mathbb{E}_{z_{<t}, u_{<t}}\left[\nabla F_S(x_t)^\top \nabla F_\lambda(x_t)\right],
\end{aligned}$$

where we use Lemma C.2 for $F_\lambda(x) = \mathbb{E}_v[F_S(x+\lambda v)]$ with v uniformly sampled from $\sqrt{d}\mathbb{B}^d$. Rearranging terms, we thus obtain that

$$\begin{aligned}
\mathbb{E}_t\left[\nabla F_S(x_t)^\top G_\lambda(x_t)\,\middle|\,Q_t\right] &= \frac{\mathbb{E}_{z_{<t},u_{<t}}\left[\nabla F_S(x_t)^\top \nabla F_\lambda(x_t)\right]}{\mathbb{P}(Q_t)} - \frac{\mathbb{E}_t\left[\nabla F_S(x_t)^\top G_\lambda(x_t)\,\middle|\,\bar{Q}_t\right]\mathbb{P}(\bar{Q}_t)}{\mathbb{P}(Q_t)} \\
&= \frac{\mathbb{E}_{z_{<t},u_{<t}}\|\nabla F_S(x_t)\|^2}{2\,\mathbb{P}(Q_t)} + \frac{\mathbb{E}_{z_{<t},u_{<t}}\|\nabla F_\lambda(x_t)\|^2}{2\,\mathbb{P}(Q_t)} - \frac{\mathbb{E}_{z_{<t},u_{<t}}\|\nabla F_S(x_t)-\nabla F_\lambda(x_t)\|^2}{2\,\mathbb{P}(Q_t)} \\
&\quad - \frac{\mathbb{E}_t\left[\nabla F_S(x_t)^\top G_\lambda(x_t)\,\middle|\,\bar{Q}_t\right]\mathbb{P}(\bar{Q}_t)}{\mathbb{P}(Q_t)} \\
&\geq \frac{\mathbb{E}_{z_{<t},u_{<t}}\|\nabla F_S(x_t)\|^2}{2\,\mathbb{P}(Q_t)} - \frac{\ell^2\lambda^2 d^3}{8\,\mathbb{P}(Q_t)} - \frac{\mathbb{E}_t\left[\nabla F_S(x_t)^\top G_\lambda(x_t)\,\middle|\,\bar{Q}_t\right]\mathbb{P}(\bar{Q}_t)}{\mathbb{P}(Q_t)},
\end{aligned}$$

where we apply (ii) in Lemma C.2. Assumption 3.5 implies that $F_S(x)$ is also Lipschitz, and thus

$$\begin{aligned}
\nabla F_S(x_t)^\top G_\lambda(x_t) &\leq \|\nabla F_S(x_t)\|\|G_\lambda(x_t)\| \\
&\leq L^2\|u_t\|^2 \\
&= L^2 d.
\end{aligned}$$

As a result, we obtain that

$$\mathbb{E}_t\left[\nabla F_S(x_t)^\top \hat{G}_\lambda(x_t)\,\middle|\,Q_t\right] \geq \frac{\mathbb{E}_{z_{<t},u_{<t}}\|\nabla F_S(x_t)\|^2}{2\,\mathbb{P}(Q_t)} - \frac{\ell^2\lambda^2 d^3}{8\,\mathbb{P}(Q_t)} - \frac{L^2 d\,\mathbb{P}(\bar{Q}_t)}{\mathbb{P}(Q_t)}. \tag{21}$$

Plugging (21), (19) and (20) back into (17), we obtain that

$$\begin{aligned}
\mathbb{E}_{z_{<(t+1)},u_{<(t+1)}}[F_S(x_{t+1})|Q_t] \leq \mathbb{E}_{z_{<t},u_{<t}}[F_S(x_t)|Q_t] - \frac{\alpha}{2}(1-2(r+2)\ell\alpha)\frac{\mathbb{E}_{z_{<t},u_{<t}}\|\nabla F_S(x_t)\|^2}{\mathbb{P}(Q_t)} + \frac{\ell\alpha^2\, r\sigma^2}{2\,\mathbb{P}(Q_t)} \\
+ \frac{\ell^2\alpha(d+2\ell\alpha r)\lambda^2 d^2}{8\,\mathbb{P}(Q_t)} + \frac{\alpha L^2 d\,\mathbb{P}(\bar{Q}_t)}{\mathbb{P}(Q_t)}.
\end{aligned} \tag{22}$$

Choosing $\alpha = 1/(4\ell(r+2))$ such that $1-2(r+2)\ell\alpha = 1/2$ and $2\ell\alpha r < 1 \leq d$, we have that

$$\begin{aligned}
\mathbb{E}_{z_{<t},u_{<t}}\|\nabla F_S(x_t)\|^2 &\leq \frac{4\,\mathbb{E}_{z_{<(t+1)},u_{<(t+1)}}[F_S(x_t)-F_S(x_{t+1})|Q_t]\mathbb{P}(Q_t)}{\alpha} + 2\ell\alpha\, r\sigma^2 + \ell^2 d^3\lambda^2 + 4L^2 d\,\mathbb{P}(\bar{Q}_t) \\
&\leq \frac{4\,\mathbb{E}_{z_{<(t+1)},u_{<(t+1)}}[F_S(x_t)-F_S(x_{t+1})|Q_t]\mathbb{P}(Q_t)}{\alpha} + 2\ell\alpha\, r\sigma^2 + \ell^2 d^3\lambda^2 + 4L^2 d\,\mathbb{P}(\bar{Q}).
\end{aligned}$$

Recall Q_t is the event that clipping does not happen at iteration t and Q is the event that clipping does not happen for every iteration. By the law of total probability and the assumption that $|F_S(x_t)| \leq B$ for every t, we have that

$$\begin{aligned}
\mathbb{E}_{z_{<(t+1)},u_{<(t+1)}}[F_S(x_t)-F_S(x_{t+1})|Q_t]\mathbb{P}(Q_t) &= \mathbb{E}_{z_{<T},u_{<T}}[F_S(x_t)-F_S(x_{t+1})|Q_t]\mathbb{P}(Q_t) \\
&= \mathbb{E}_{z_{<T},u_{<T}}\left[F_S(x_t)-F_S(x_{t+1})\middle|Q_t\cap Q\right]\mathbb{P}(Q_t\cap Q) \\
&\quad + \mathbb{E}_{z_{<T},u_{<T}}\left[F_S(x_t)-F_S(x_{t+1})\middle|Q_t\cap \bar{Q}\right]\mathbb{P}(Q_t\cap \bar{Q}) \\
&\leq \mathbb{E}_{z_{<T},u_{<T}}[F_S(x_t)-F_S(x_{t+1})|Q]\mathbb{P}(Q) + 2B\,\mathbb{P}(\bar{Q}).
\end{aligned}$$

As a result, we have that

$$\mathbb{E}_{z_{<t},u_{<t}}\|\nabla F_S(x_t)\|^2 \leq \frac{4\,\mathbb{E}_{z_{<T},u_{<T}}[F_S(x_t)-F_S(x_{t+1})|Q]\mathbb{P}(Q)}{\alpha} + 2\ell\alpha\, r\sigma^2 + \ell^2 d^3\lambda^2 + \left(4L^2 d + \frac{8B}{\alpha}\right)\mathbb{P}(\bar{Q}). \tag{23}$$

Taking expectation with respect to all randomness, i.e., $\mathbb{E} = \mathbb{E}_{z_{<T}, u_{<T}}$, summing up from $t = 0$ to $T-1$, and dividing both sides by T, we have that

$$\begin{aligned}
\mathbb{E}\|\nabla F_S(x_\tau)\|^2 &= \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{z_{<t}, u_{<t}} \|\nabla F_S(x_t)\|^2 \\
&\le \frac{4\,\mathbb{E}[F_S(x_0) - F_S(x_T)|Q]\mathbb{P}(Q)}{\alpha T} + \frac{64\ell C^2\,\alpha T\, r\log(e + (\varepsilon/\delta))}{n^2\varepsilon^2} + \ell^2 d^3\lambda^2 \\
&\quad + 8\sqrt{2\pi}\, n\, T(L^2 d + 8\ell B(r+2)) \exp\left(-\frac{C_0^2}{8L^2}\right) \\
&\le \left(64\ell[F_S(x_0) - F_S^*] + 4C^2\right)\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon} + \ell^2 d^3\lambda^2 \\
&\quad + \frac{2\sqrt{2\pi}\, n^2\varepsilon(r+2)(L^2 d + 8\ell B(r+2))}{\sqrt{r\log(e + (\varepsilon/\delta))}} \exp\left(-\frac{C_0^2}{8L^2}\right),
\end{aligned}$$

with the choice of parameters to be

$$\alpha T = \frac{n\varepsilon}{16\ell\sqrt{r\log(e + (\varepsilon/\delta))}}, \quad \alpha = \frac{1}{4\ell(r+2)}, \quad T = \frac{n(r+2)\varepsilon}{4\sqrt{r\log(e + (\varepsilon/\delta))}}.$$

When selecting $\lambda \le 2(\sqrt{2}-1)C_0/(\ell d)$, we can set $C = \sqrt{2}C_0$ such that $C \ge C_0 + \ell\lambda d/2$ is satisfied. If C_0 and λ further satisfy the conditions that

$$C_0^2 = 8L^2\,\log\left(\frac{2\sqrt{2\pi}\, n^3\varepsilon^2(r+2)(d + 8\ell B(r+2)/L^2)}{r\log(e + (\varepsilon/\delta))}\right), \quad \lambda \le \frac{L}{\ell d^{3/2}}\left(\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}\right)^{1/2},$$

we can then obtain that

$$\begin{aligned}
&\mathbb{E}\|\nabla F_S(x_\tau)\|^2 \\
&\quad \le \left(64\,\ell[F_S(x_0) - F_S^*] + 4C^2 + 2L^2\right)\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon} \\
&\quad = \left(64\,\ell[F_S(x_0) - F_S^*] + 64\,L^2\,\log\left(\frac{2\sqrt{2\pi}\, n^3\varepsilon^2(r+2)(d + 8\ell B(r+2)/L^2)}{r\log(e + (\varepsilon/\delta))}\right) + 2L^2\right)\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}.
\end{aligned}$$

We conclude that the clipping threshold C and smoothing parameter λ should satisfy that

$$\begin{aligned}
C &= 4L\sqrt{\log\left(\frac{2\sqrt{2\pi}\, n^3\varepsilon^2(r+2)(d + 8\ell B(r+2)/L^2)}{r\log(e + (\varepsilon/\delta))}\right)}, \\
\lambda &\le \frac{L}{\ell d}\min\left\{4(2-\sqrt{2})\sqrt{\log\left(\frac{2\sqrt{2\pi}\, n^3\varepsilon^2(r+2)(d + 8\ell B(r+2)/L^2)}{r\log(e + (\varepsilon/\delta))}\right)}, \frac{1}{\sqrt{d}}\left(\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}\right)^{1/2}\right\}.
\end{aligned}$$

The total number of zeroth-order gradient computations is $nT = n^2(r+2)\varepsilon/(4\sqrt{r\log(e + (\varepsilon/\delta))})$.

F Extension to the PL Setting

Assumption F.1. The average loss $F_S(x)$ satisfies the PL inequality with parameter $\mu > 0$. That is, it holds that $\forall x \in \mathbb{R}^d$,

$$\|\nabla F_S(x)\|^2 \ge 2\mu(F_S(x) - F_S^*).$$

Corollary F.2. *Under the same setting of Theorem 1, when Assumption F.1 is also met, let $\kappa = \ell/\mu$ be the condition number, the last iterate of Algorithm 1 satisfies that*

$$\mathbb{E}[F_S(x_T) - F_S^*] \leq \left(\ell(F_S(x_0) - F_S^*) + 64L^2\kappa \log\left(\frac{n^2\varepsilon^2}{\kappa\, d^3 \log(e + (\varepsilon/\delta))}\right) + 2L^2\right) \frac{d^3 \log(e + (\varepsilon/\delta))}{\mu n^2 \varepsilon^2},$$

with the choice of parameters

$$\alpha = \frac{1}{4\ell d}, \quad T = 8\,\kappa\, d\, \log\left(\frac{n^2\varepsilon^2}{\kappa\, d^3 \log(e + (\varepsilon/\delta))}\right), \quad \lambda \leq \frac{2L}{\ell} \frac{\sqrt{\log(e + (\varepsilon/\delta))}}{n\varepsilon}, \quad C = Ld.$$

The total number of zeroth-order gradient computations is $nT = \tilde{\mathcal{O}}(nd\kappa)$.

Proof. Starting from (12) in the proof of Theorem 1, with the choice that $\alpha = 1/(4\ell d)$, we have that

$$\begin{aligned}
\mathbb{E}_{z_t, u_t}[F_S(x_{t+1})] &\leq F_S(x_t) - \frac{\alpha}{4}\|F_S(x_t)\|^2 + \frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, d\sigma^2 \\
&\leq F_S(x_t) - \frac{\mu\alpha}{2}(F_S(x_t) - F_S^*) + \frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, d\sigma^2.
\end{aligned}$$

This gives the recursion that

$$\mathbb{E}[F_S(x_{t+1}) - F_S^*] \leq \left(1 - \frac{\mu\alpha}{2}\right) \mathbb{E}[F_S(x_t) - F_S^*] + \frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, d\sigma^2.$$

Resolving the recursion, we obtain that

$$\begin{aligned}
\mathbb{E}[F_S(x_T) - F_S^*] &\leq \left(1 - \frac{\mu\alpha}{2}\right)^T (F_S(x_0) - F_S^*) + \left(\frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, d\sigma^2\right)\left(\left(1 - \frac{\mu\alpha}{2}\right)^{T-1} + \cdots + \left(1 - \frac{\mu\alpha}{2}\right) + 1\right) \\
&\leq \exp\left(-\frac{\mu\alpha T}{2}\right)(F_S(x_0) - F_S^*) + \frac{\ell^2\lambda^2 d^3}{2\mu} + \frac{\ell\alpha\, d\sigma^2}{\mu} \\
&= \exp\left(-\frac{\mu\alpha T}{2}\right)(F_S(x_0) - F_S^*) + \frac{32\ell L^2\, \alpha T\, d^3 \log(e + (\varepsilon/\delta))}{\mu n^2\varepsilon^2} + \frac{\ell^2\lambda^2 d^3}{2\mu} \\
&= \left(\ell(F_S(x_0) - F_S^*) + 64L^2\kappa \log\left(\frac{n^2\varepsilon^2}{\kappa\, d^3 \log(e + (\varepsilon/\delta))}\right) + 2L^2\right) \frac{d^3 \log(e + (\varepsilon/\delta))}{\mu n^2 \varepsilon^2},
\end{aligned}$$

with the choice of parameters

$$\alpha T = \frac{2}{\mu} \log\left(\frac{n^2\varepsilon^2}{\kappa\, d^3 \log(e + (\varepsilon/\delta))}\right), \quad \lambda \leq \frac{2L}{\ell} \frac{\sqrt{\log(e + (\varepsilon/\delta))}}{n\varepsilon}.$$

The total number of iteration is $T = \tilde{\mathcal{O}}(\kappa d)$. □

Corollary F.3. *Under the same setting of Theorem 2, when Assumption F.1 is also met, let $\kappa = \ell/\mu$ be the condition number, the last iterate of Algorithm 1 satisfies that*

$$\mathbb{E}[F_S(x_T) - F_S^*] \leq \left(\ell(F_S(x_0) - F_S^*) + 64L^2\kappa \log\left(\frac{n^2\varepsilon^2}{\kappa\, r d^2 \log(e + (\varepsilon/\delta))}\right) + 2L^2\right) \frac{r d^2 \log(e + (\varepsilon/\delta))}{\mu n^2 \varepsilon^2},$$

with the choice of parameters

$$\alpha = \frac{1}{4\ell(r+2)}, \quad T = 8\,\kappa\,(r+2)\, \log\left(\frac{n^2\varepsilon^2}{\kappa\, r d^2 \log(e + (\varepsilon/\delta))}\right), \quad \lambda \leq \frac{2L}{\ell\sqrt{d}} \frac{\sqrt{r \log(e + (\varepsilon/\delta))}}{n\varepsilon}, \quad C = Ld.$$

The total number of zeroth-order gradient computations is $nT = \tilde{\mathcal{O}}(nr\kappa)$.

Proof. Starting from (15) in the proof of Theorem 2, with the choice that $\alpha = 1/(4\ell(r+2))$, we have that

$$\begin{aligned}\mathbb{E}_{z_t,u_t}[F_S(x_{t+1})] &\le F_S(x_t) - \frac{\alpha}{4}\|F_S(x_t)\|^2 + \frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, r\sigma^2 \\ &\le F_S(x_t) - \frac{\mu\alpha}{2}(F_S(x_t) - F_S^*) + \frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, r\sigma^2.\end{aligned}$$

This gives the recursion that

$$\mathbb{E}[F_S(x_{t+1}) - F_S^*] \le \left(1 - \frac{\mu\alpha}{2}\right)\mathbb{E}[F_S(x_t) - F_S^*] + \frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, r\sigma^2.$$

Resolving the recursion, we obtain that

$$\begin{aligned}\mathbb{E}[F_S(x_T) - F_S^*] &\le \left(1 - \frac{\mu\alpha}{2}\right)^T (F_S(x_0) - F_S^*) + \left(\frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, r\sigma^2\right)\left(\left(1 - \frac{\mu\alpha}{2}\right)^{T-1} + \cdots + \left(1 - \frac{\mu\alpha}{2}\right) + 1\right) \\ &\le \exp\left(-\frac{\mu\alpha T}{2}\right)(F_S(x_0) - F_S^*) + \frac{\ell^2\lambda^2 d^3}{2\mu} + \frac{\ell\alpha\, r\sigma^2}{\mu} \\ &= \exp\left(-\frac{\mu\alpha T}{2}\right)(F_S(x_0) - F_S^*) + \frac{32\ell L^2\,\alpha T\, rd^2\log(e + (\varepsilon/\delta))}{\mu n^2\varepsilon^2} + \frac{\ell^2\lambda^2 d^3}{2\mu} \\ &= \left(\ell(F_S(x_0) - F_S^*) + 64L^2\kappa\,\log\left(\frac{n^2\varepsilon^2}{\kappa\, rd^2\log(e + (\varepsilon/\delta))}\right) + 2L^2\right)\frac{rd^2\log(e + (\varepsilon/\delta))}{\mu n^2\varepsilon^2},\end{aligned}$$

with the choice of parameters

$$\alpha T = \frac{2}{\mu}\log\left(\frac{n^2\varepsilon^2}{\kappa\, rd^2\log(e + (\varepsilon/\delta))}\right), \quad \lambda \le \frac{2L}{\ell\sqrt{d}}\,\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}.$$

The total number of iteration is $T = \tilde{\mathcal{O}}(\kappa r)$. □

Corollary F.4. *Under the same setting of Theorem 3, when Assumption F.1 is also met, let $\kappa = \ell/\mu$ be the condition number, suppose $\max_{0\le t\le T}|F_S(x_t)| \le B$ and $|F_S^*| \le B$, the last iterate of Algorithm 2 satisfies that*

$$\mathbb{E}[F_S(x_T) - F_S^*] \le \left(\ell(F_S(x_0) - F_S^*) + \log\left(\frac{n^2\varepsilon^2}{\kappa\, r\log(e + (\varepsilon/\delta))}\right)\left(L^2 + 16\tilde{L}^2\kappa\right) + 2L^2\right)\frac{r\log(e + (\varepsilon/\delta))}{\mu n^2\varepsilon^2},$$

where we define

$$\tilde{L}^2 = 64L^2\log\left(\frac{32\sqrt{2\pi}\,\kappa\, n^3\varepsilon^2(r+2)(d + (8\ell(r+2) + \mu)B/L^2)}{r\log(e + (\varepsilon/\delta))}\right),$$

and choose the parameters to be

$$\begin{aligned}&\alpha = \frac{1}{4\ell(r+2)}, \quad T = 8\,\kappa\,(r+2)\,\log\left(\frac{n^2\varepsilon^2}{\kappa\, r\log(e + (\varepsilon/\delta))}\right), \quad C = \frac{\tilde{L}}{2}, \\ &\lambda \le \frac{1}{2\ell d}\,\min\left\{(2 - \sqrt{2})\tilde{L},\ \frac{4L}{\sqrt{d}}\,\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}\right\}.\end{aligned}$$

The total number of zeroth-order gradient computations is $nT = \tilde{\mathcal{O}}(nr\kappa)$.

Remark F.5. A more precise expression of our theoretical results, including Theorems 1, 2, and 3 and their corresponding Corollaries F.2, F.3, and F.4, is to cover cases where T may be less than 1. Considering Theorem 3 as an example, a more accurate statement is

$$T = \max\left\{\frac{n(r+2)\varepsilon}{4\sqrt{r\log(e + (\varepsilon/\delta))}}, 1\right\}, \quad \mathbb{E}[\|\nabla F_S(x_\tau)\|^2] \le \min\left\{\tilde{\mathcal{O}}\left(\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}\right), L^2\right\}.$$

For the sake of clarity and simplicity in presentation, this detail is omitted in the main results.

Proof. Starting from (23) in the proof of Theorem 3 with the choice $\alpha = 1/(4\ell(r+2))$ and using Assumption F.1 such that

$$\begin{aligned}\mathbb{E}\|\nabla F_S(x_t)\|^2 &\geq 2\mu\,\mathbb{E}[F_S(x_t) - F_S^*] \\ &= 2\mu\,\mathbb{E}[F_S(x_t) - F_S^*|Q]\,\mathbb{P}(Q) + 2\mu\,\mathbb{E}[F_S(x_t) - F_S^*|\bar{Q}]\,\mathbb{P}(\bar{Q}) \\ &\geq 2\mu\,\mathbb{E}[F_S(x_t) - F_S^*|Q]\,\mathbb{P}(Q),\end{aligned}$$

we have the recursion that

$$\mathbb{E}[F_S(x_{t+1}) - F_S^*|Q]\mathbb{P}(Q) \leq \left(1 - \frac{\mu\alpha}{2}\right)\mathbb{E}[F_S(x_t) - F_S^*|Q]\mathbb{P}(Q) + \frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, r\sigma^2 + (L^2 d\alpha + 2B)\mathbb{P}(\bar{Q}).$$

Resolving the recursion, we obtain that

$$\begin{aligned}\mathbb{E}[F_S(x_T) - F_S^*|Q]\mathbb{P}(Q) &\leq \left(1 - \frac{\mu\alpha}{2}\right)^T (F_S(x_0) - F_S^*) + \frac{2}{\mu\alpha}\left(\frac{\alpha}{4}\,\ell^2\lambda^2 d^3 + \frac{\ell}{2}\alpha^2\, r\sigma^2 + (L^2 d\alpha + 2B)\mathbb{P}(\bar{Q})\right) \\ &\leq \exp\left(-\frac{\mu\alpha T}{2}\right)(F_S(x_0) - F_S^*) + \frac{\ell^2\lambda^2 d^3}{2\mu} + \frac{\ell\alpha\, r\sigma^2}{\mu} + \frac{(2L^2 d + 4B/\alpha)\mathbb{P}(\bar{Q})}{\mu}.\end{aligned}$$

Since the event Q happens with high probability, the above results can be refined to

$$\begin{aligned}\mathbb{E}[F_S(x_T) - F_S^*] &= \mathbb{E}[F_S(x_T) - F_S^*|Q]\mathbb{P}(Q) + \mathbb{E}[F_S(x_T) - F_S^*|\bar{Q}]\mathbb{P}(\bar{Q}) \\ &\leq \mathbb{E}[F_S(x_T) - F_S^*|Q]\mathbb{P}(Q) + 2B\;\mathbb{P}(\bar{Q}).\end{aligned}$$

Therefore, we can obtain that

$$\begin{aligned}\mathbb{E}[F_S(x_T) - F_S^*] &\leq \exp\left(-\frac{\mu\alpha T}{2}\right)(F_S(x_0) - F_S^*) + \frac{32\ell C^2\,\alpha T\, r\log(e + (\varepsilon/\delta))}{\mu n^2\varepsilon^2} \\ &\quad + \frac{4\sqrt{2\pi}nT(L^2 d + 2B/\alpha + B\mu)}{\mu}\exp\left(-\frac{C_0^2}{8L^2}\right) + \frac{\ell^2\lambda^2 d^3}{2\mu} \\ &= \left(\ell(F_S(x_0) - F_S^*) + L^2\,\log\left(\frac{n^2\varepsilon^2}{\kappa\, r\log(e + (\varepsilon/\delta))}\right)\right)\frac{r\log(e + (\varepsilon/\delta))}{\mu n^2\varepsilon^2} \\ &\quad + \frac{32\ell C^2\,\alpha T\, r\log(e + (\varepsilon/\delta))}{\mu n^2\varepsilon^2} + \frac{\ell^2\lambda^2 d^3}{2\mu},\end{aligned}$$

with the choice of parameters

$$\alpha T = \frac{2}{\mu}\log\left(\frac{n^2\varepsilon^2}{\kappa\, r\log(e + (\varepsilon/\delta))}\right), \quad C_0^2 = 8L^2\,\log\left(\frac{32\sqrt{2\pi}\,\kappa\, n^3\varepsilon^2(r+2)(d + (8\ell(r+2) + \mu)B/L^2)}{r\log(e + (\varepsilon/\delta))}\right).$$

When selecting λ to be

$$\lambda \leq \min\left\{\frac{2(\sqrt{2} - 1)C_0}{\ell d},\; \frac{2L}{\ell d^{3/2}}\frac{\sqrt{r\log(e + (\varepsilon/\delta))}}{n\varepsilon}\right\},$$

we can set $C = \sqrt{2}C_0$ such that $C \geq C_0 + \ell\lambda d/2$ is satisfied, and thus

$$\mathbb{E}[F_S(x_T) - F_S^*] \leq \left(\ell(F_S(x_0) - F_S^*) + \log\left(\frac{n^2\varepsilon^2}{\kappa\, r\log(e + (\varepsilon/\delta))}\right)\left(L^2 + 16\tilde{L}^2\kappa\right) + 2L^2\right)\frac{r\log(e + (\varepsilon/\delta))}{\mu n^2\varepsilon^2},$$

where we define

$$\tilde{L}^2 = 64L^2\log\left(\frac{32\sqrt{2\pi}\,\kappa\, n^3\varepsilon^2(r+2)(d + (8\ell(r+2) + \mu)B/L^2)}{r\log(e + (\varepsilon/\delta))}\right).$$

The total number of iteration is $T = \tilde{\mathcal{O}}(\kappa r)$. □