Private Zeroth-Order Optimization with Public Data

Xuchen Gong [1] **Tian Li** [1]

Abstract

One of the major bottlenecks for deploying popular first-order differentially private (DP) machine learning algorithms (e.g., DP-SGD) lies in their high computation and memory cost, despite the existence of optimized implementations. Zeroth-order methods have promise in mitigating the overhead, as they leverage function evaluations to approximate the gradients, hence significantly easier to privatize. While recent works have explored zeroth-order approaches in both private and non-private settings, they still suffer from relatively low utilities compared with DP-SGD and limited application domains. In this work, we propose to leverage public information to guide and improve gradient approximation of private zeroth-order algorithms. We explore a suite of public-data-assisted zeroth-order optimizers (PAZO) with minimal overhead. We provide theoretical analyses of the PAZO framework under an assumption of the similarity between public and private data. Empirically, we demonstrate that PAZO achieves stronger privacy/utility tradeoffs across vision and text tasks in both pre-training and fine-tuning regimes, outperforming the best first-order baselines (with public gradients) especially in highly private regimes, while offering up to $16\times$ runtime speedup.

1. Introduction

Differentially private (DP) offers a widely-used framework to protect sensitive information so that adversaries cannot infer if any user/sample participates in the computation. When applied to machine learning tasks, popular DP algorithms based on privatizing first-order gradients (such as DP-SGD (Abadi et al., 2016)) fundamentally rely on per-sample gradient clipping, which can be computationally expensive and impractical in large-scale settings. While there exist optimized implementations of DP-SGD, they are limited in their generality to handle all model architectures and often incur other overheads, such as trading memory for computation (Subramani et al., 2021; Beltran et al.).

To tackle this, zeroth-order optimization offers an attractive alternative for DP training, as it leverages function queries (one-dimensional scalar values) to approximate the gradients and is hence inherently amenable to privatization (Duchi et al., 2015; Kiefer & Wolfowitz, 1952). However, randomly searching in a potentially high-dimensional space based on function query feedback can be rather inefficient (Duchi et al., 2015). Prior work has demonstrated competitive performance of (private) zeroth-order methods only in the limited context of language model prompt tuning (Malladi et al., 2023; Tang et al., 2024; Zhang et al., 2023; Ma & Huang; Zhang et al., 2024b) or under extreme sparsity (Chen et al., 2023). In addition, there is still a utility gap between private zeroth-order and first-order approaches on challenging tasks (Zhang et al., 2023).

In this work, we aim to narrow the gap between zeroth-order and first-order methods in private training leveraging public data. Zeroth-order outputs are high-variance estimators of the first-order gradients and suffer from slow convergence in terms of the total number of iterations. However, there usually exists public data that is exempt from privatization, whose batch gradient provides informative guidance and introduces minimal computational overhead. We thus introduce PAZO, a suite of zeroth-order DP optimizers that leverage a small amount of public data with similar distributions as private data and their first-order gradients to guide or augment the zeroth-order outputs. In particular, we explore (1) PAZO-M, a mix (convex combination) of private zeroth-order estimates and public first-order gradients, (2) PAZO-P, constraining the sampling of random directions in the public gradient subspace, and (3) PAZO-S, selecting the best public gradient based on function queries on private data. When designing PAZO, we ensure that privatization only operates on top of function evaluations to preserve the efficiency of zeroth-order approaches, while still satisfying desired privacy guarantees.

Unlike recent zeroth-order work that mostly focuses on language model prompt tuning, we cover both image and text

[1] University of Chicago. Correspondence to: Xuchen Gong <xuchengo@uchicago.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

domains, and both pre-training and fine-tuning scenarios. We show that without access to public data, DP zeroth-order methods may underperform DP first-order approaches (e.g., DP-SGD), whereas even modest amounts of public data can significantly close the gap, especially in highly private regimes. In particular, the best zeroth-order with public data method can match or even outperform the best first-order method with public data, while being significantly faster to train. Our results highlight the broader potential of zeroth-order methods for DP training with public data: enabling improved privacy/utility tradeoffs, applicability across diverse domains, and achieving up to 16× speedup compared to traditional first-order methods. Our contributions are as follows:

1. **Algorithm design.** We propose the first set of private zeroth-order optimization algorithms (PAZO-{M,P,S}) augmented with public data (gradients) to construct better gradient estimates in a more constrained space. PAZO helps close the gap between zeroth- and first-order methods in the settings where zeroth-order approaches underperform first-order ones.

2. **Theoretical analysis.** We present the privacy and utility guarantees for each method. We verify that our worst-case convergence rate matches that of previous work, and a proper choice of hyperparameter, which depends on the quality of public data, gives us improved convergence.

3. **Empirical validation.** We evaluate our methods on both image and text domains and in both pre-training and fine-tuning scenarios. We find that zeroth-order methods are robust across various privacy budgets whereas first-order methods are sensitive. Our methods consistently have superior privacy/utility tradeoffs and outperform the best public-augmented first-order method in highly privacy regimes, while achieving up to 16× speedup.

2. Preliminaries

Differential privacy. In this work, we focus on the classic definition of sample-level DP (Abadi et al., 2016; Dwork et al., 2006).

Definition 2.1 (Differential privacy (Dwork et al., 2006)). *A randomized algorithm $\mathcal{M}$ is (ε, δ)-differentially private if for all neighboring datasets D, D' differing by one element, and every possible subset of outputs O,*

$$\Pr(\mathcal{M}(D){\in}O){\leq}e^{\varepsilon}\Pr(\mathcal{M}(D'){\in}O){+}\delta.$$

We follow the classic DP model where the neighboring datasets D and D' differ by adding/removing one training sample. Typically, noise is added to ensure DP scales with the model dimensions, resulting in degraded and unusable model utilities (Chaudhuri et al., 2011). Extensive prior research has been proposed to improve privacy/utility tradeoffs, including increasing the batch-size (McMahan et al., 2017; Sander et al., 2024), using public or side information (Li et al., 2022; Asi et al., 2021; Li et al., 2021), and reducing the dimensionality of gradients (Zhou et al., 2020). Another bottleneck of deploying DP algorithms at scale lies in the computation (or memory) cost (Subramani et al., 2021). In this work, we propose to mix zeroth-order (on sensitive private data) and first-order oracles (on public data) to mitigate these two challenges at once.

Zeroth-order optimization. Zeroth-order approaches use (stochastic) function queries to estimate the true gradients. They are particularly suitable for applications where gradient information is difficult to obtain, such as adversarial attacks and defenses (Chen et al., 2017; Ilyas et al., 2018; Verma et al., 2023), hyperparameter tuning (Gu et al., 2021), and data-driven science workloads (Hoffman et al., 2022). One fundamental challenge of zeroth-order methods is their need for a large number of function queries to reduce the variance of the estimate (e.g., Duchi et al., 2015). Existing work has explored various techniques to improve the estimate, such as incorporating the previous estimated gradient directions (Meier et al., 2019) and sparsifying gradients (Chen et al., 2023). This work focuses on private training, and our proposed technique can be combined with these prior methods. Given the current model parameter $x \in \mathbb{R}^d$ and loss function $f : \mathbb{R}^d \to \mathbb{R}$, the widely used two-point zeroth-order gradient estimator (Duchi et al., 2015), involves two evaluations of function values:

$$g_{\lambda}(x;\xi_i){:=}\frac{f(x{+}\lambda u;\xi_i){-}f(x{-}\lambda u;\xi_i)}{2\lambda}u, \tag{1}$$

where ξ_i is a randomly sampled training data point, $u{\in}\mathbb{R}^d$ is uniformly sampled from the Euclidean sphere $\sqrt{d}\mathbb{S}^{d-1}$, and $\lambda{>}0$ is the smoothing parameter. Let v be uniformly sampled from the Euclidean ball $\sqrt{d}\mathbb{B}^d{=}\{x{\in}\mathbb{R}^d|\|x\|{\leq}\sqrt{d}\}$. Define the smoothed version of $f(\cdot)$ as $f_{\lambda}(x){:=}\mathbb{E}_v[f(x{+}\lambda v)]$. We have that (i) $f_{\lambda}(x)$ is differentiable and (ii) $\mathbb{E}_u[g_{\lambda}(x;\xi_i)]{=}\nabla f_{\lambda}(x)$. It indicates that by using the zeroth-order gradient estimator, we are asymptotically optimizing a smoothed version of the original objective $f(x)$, where the smoother is a ball with radius $\lambda\sqrt{d}$.

Differentially private zeroth-order optimization. The desired (private) gradients are expensive to obtain in DP training, because gradients have to be generated and privatized at a granularity of each sample as opposed to each mini-batch. Therefore, recent work has considered privatizing zeroth-order algorithms (Zhang et al., 2023; Tang et al., 2024; Liu et al., 2024; Zhang et al., 2024a) by first clipping the function queries and then adding proper Gaussian noise. Specifically, based on the non-private two-point estimator on one sample (Eq. (1)), prior work uses the privatized

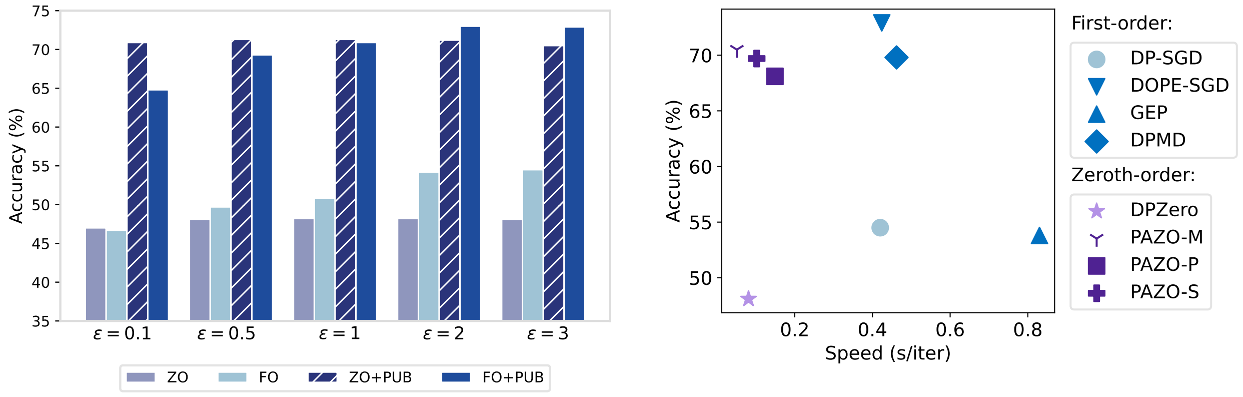


Figure 1: Results of CIFAR-10 with NFResNet18 trained from scratch. *Left:* Zeroth-order methods demonstrate consistent accuracies under various privacy budgets compared with the best first-order method with public data. *Right:* Proposed zeroth-order approaches are more accurate than vanilla DPZero, and significantly more efficient than all the public data augmented first-order baselines. The results are under the privacy budget $\varepsilon = 3$.

update rule $\tilde{g}_\lambda(x;B)$ defined by

$$\left(\frac{1}{b}\sum_{\xi_i\in B}\text{clip}_C\left(\frac{f(x+\lambda u;\xi_i)-f(x-\lambda u;\xi_i)}{2\lambda}\right)+z\right)u, \quad (2)$$

where $b=|B|$ is batch-size, $z\sim\frac{1}{b}\mathcal{N}(0,C^2\sigma^2)$ is privacy noise, and u is sampled uniformly from the sphere $\sqrt{d}\mathbb{S}^{d-1}$. It is possible to query the raw data multiple times per iteration by sampling multiple u's with more noise added (Section 3). However, prior work mostly focuses on language model prompt tuning, and there still exists a big performance gap between zeroth- and first-order methods. In PAZO, we use public information to guide the gradient estimate on private data, as discussed in the next section.

3. PAZO: Public-Data-Assisted Zeroth-Order Optimization

Given zeroth-order oracles on private data and first-order oracles on public data, we aim to blend public gradients as inductive bias into the private zeroth-order framework to improve privacy/utility tradeoffs, while retaining the efficiency benefits of vanilla zeroth-order updates. In this section, we propose three approaches for using this public prior, which significantly outperform baselines without public data and result in competitive/superior performance relative to DP-SGD with public data. We analyze the convergence properties in Section 4.

3.1. PAZO-M: Mixing Zeroth-Order Estimates and First-Order Gradients

PAZO-M linearly combines the public gradient with the private two-point estimator (Eq. (2)), summarized in Algorithm 1 below. At each iteration t, we sample a public batch, obtain its batch gradient, and mix it with the private two-point gradient estimate. We run private two-point estimation q times to reduce its variance. Since we query the same raw private mini-batch q times, we need to add more privacy noise (q times more variance) to ensure the same DP as if querying once.

Since the norm of two-point gradient estimates is approximately d times that of the true private gradient, it is important to align their norm so that the tuning of the mixing coefficient is independent of the problem. To achieve this, we sample u uniformly from the sphere $r\mathbb{S}^{d-1}$ with radius $r=d^{\frac{1}{4}}$ so that $\mathbb{E}_{u_t}[\|g_\lambda(x)\|^2]\approx\|\nabla f(x)\|^2$, whose proof detailed in Appendix A.

One can adjust the mixing coefficient α to adjust the weight put on the public gradient. Although α is an additional hyperparameter, as we show in our experiments (Section 5), PAZO-M is robust to a wide range of $\alpha\in(0,1)$ and public batch-size b', as long as the L_2 norms of g_{pub} and $\tilde{g}/q$ are at the same scale.

Despite its simplicity, PAZO-M demonstrates competitive performance among all three PAZO variants (Section 5). While prior work has explored mixing gradients and zeroth-order estimates for memory efficiency in non-private settings (Li et al., 2024), PAZO-M differs from this work in terms of the effective optimization objectives, bias-variance tradeoffs, the analysis, and application settings.

3.2. PAZO-P: Sampling in Public Gradient Subspace

Recall that the two-point estimator samples perturbations u in the sphere $\sqrt{d}\mathbb{S}^{d-1}$, while such random exploration along the directions λu and $-\lambda u$ offers limited signal in terms of the real gradients. Rather, the true gradient likely lies close

Algorithm 1 PAZO-M

```
Input: $T$, noise multiplier $\sigma$, clipping threshold $C$, stepsize $\eta$, smoothing parameter $\lambda$, mixing coefficient $\alpha$, initialization $x_0 \in \mathbb{R}^d$, number of queries $q$, private and public batch-size $b$ and $b'$.
for $t=0,\cdots,T-1$ do
    Sample a batch $B$ of private training data $\{\xi_1,...,\xi_b\}$
    Sample a batch $B'$ of public data and obtain its gradient $g_{\texttt{pub}}$
    $\tilde{g} \leftarrow 0^d$
    for each of the $q$ queries do
        Sample $u$ uniformly from the sphere $d^{\frac{1}{4}}\mathbb{S}^{d-1}$
        Sample $z \sim \frac{1}{b}\mathcal{N}(0,qC^2\sigma^2)$
        $f_+ \leftarrow f(x_t+\lambda u;\xi_i)$
        $f_- \leftarrow f(x_t-\lambda u;\xi_i)$
        $\tilde{g} \leftarrow \tilde{g} + \left(\frac{1}{b}\sum_{i=1}^{b}\text{clip}_C\left(\frac{f_+-f_-}{2\lambda}\right)+z\right)u$
    end for
    $x_{t+1} \leftarrow x_t - \eta(\alpha g_{\texttt{pub}} + (1-\alpha)\tilde{g}/q)$
end for
```

Algorithm 2 PAZO-P

```
Input: Same as Algorithm 1, and number of public batches $k \ll d$
for $t=0,\cdots,T-1$ do
    Sample a batch $B$ of private training data $\{\xi_1,...,\xi_b\}$
    Sample $k$ batches of public data and obtain their (ortho)normalized gradients $\{g_1,...,g_k\}$
    $G \leftarrow [g_1,...,g_k]$, $\tilde{g} \leftarrow 0^d$
    for each of the $q$ queries do
        Sample $u$ uniformly from the sphere $\sqrt{k}\mathbb{S}^{k-1}$
        Sample $z \sim \frac{1}{b}\mathcal{N}(0,qC^2\sigma^2)$
        $f_+ \leftarrow f(x_t+\lambda Gu;\xi_i)$
        $f_- \leftarrow f(x_t-\lambda Gu;\xi_i)$
        $\tilde{g} \leftarrow \tilde{g} + \left(\frac{1}{b}\sum_{i=1}^{b}\text{clip}_C\left(\frac{f_+-f_-}{2\lambda}\right)+z\right)Gu$
    end for
    $x_{t+1} \leftarrow x_t - \eta\tilde{g}/q$
end for
```

to the space of public gradients. Based on this hypothesis, we constrain the private gradient estimates in the subspace spanned by the public gradients, and *use function queries to learn the coefficients* associated with the components of the public gradient subspace (named PAZO-P), which results in a much lower-dimensional optimization problem.

Formally, suppose we have access to k ($k \ll d$) mini-batch stochastic gradients obtained on public data and denote a concatenation of them as a matrix $G \in \mathbb{R}^{d\times k}$. Let $u \in \mathbb{R}^k$ be a random vector that is uniformly sampled from the sphere $\sqrt{k}\mathbb{S}^{k-1}$, and we propose the following updating rule based on one sample in the non-private case:

$$g_\lambda^G(x;\xi_i) := \frac{f(x+\lambda Gu;\xi_i) - f(x-\lambda Gu;\xi_i)}{2\lambda} Gu,$$

which can be interpreted as learning the coefficient $u \in \mathbb{R}^k$ to linearly combine the public gradients. Further, if we orthonormalize the columns of G, $g_\lambda^G(x;\xi_i)$ estimates the orthogonal projection of the true gradient onto the public gradient subspace when $\lambda \to 0$, i.e.,

$$\mathbb{E}_u[g_\lambda^G(x;\xi_i)] = \mathbb{E}_u[\nabla f(x)^\top GuGu] = \text{Proj}_G(\nabla f(x)).$$

We compare the visualization of sampling in the full-dimensional space and public gradient subspace in Figure 7 in the appendix. For private training, we privatize the estimates using the standard subsampled Gaussian mechanism, described in Algorithm 2.

PAZO-P is conceptually related to the idea of model soup, where extensive research has shown that a simple convex combination of the model parameters can result in a souped model that generalizes well even in out-of-distribution tasks (Wortsman et al., 2022; Croce et al., 2023). When G is not orthonormalized, PAZO-P learns the optimal convex combination via function queries privately.

Previous work proposes constraining the random search to the principal components of surrogate gradients (Maheswaranathan et al., 2019), while PAZO-P differs from theirs in the option of using non-orthonormalized G. Section 5 presents the performance of PAZO-P with orthonormalization, and the complete Table 1-4 presents the almost equally competitive performance of PAZO-P without orthonormalization.

3.3. PAZO-S: Select the Best Public Gradient

PAZO-P offers ways to better combine public gradients via zeroth-order function evaluations, while in this section, we take an alternative approach by optimizing an approximation of the problem. Note that for a convex function f, for any probability distribution $\alpha \in \Delta_k$, k public gradients $\{g_1,...,g_k\}$, and any model parameter $x \in \mathbb{R}^d$, we have that

$$\begin{aligned}\min_{\alpha\in\Delta_k} f\left(x-\eta\sum_{j=1}^{k}\alpha_j g_j\right) &\leq \min_{\alpha\in\Delta_k}\sum_{j=1}^{k}\alpha_j f(x-\eta g_j)\\ &= \min_{j\in[k]} f(x-\eta g_j),\end{aligned} \tag{3}$$

where the upper bound $\min_{j\in[k]} f(x-\eta g_j)$ can be easily optimized and privatized (as long as k is small) with access to queries of $f(\cdot)$ evaluated on private data. We propose PAZO-S, a method that selects the best public gradients based on loss values on private data, i.e., solving $\min_{j\in[k]} f(x-\eta g_j)$ (Line 6-11 in Algorithm 3). Considering the residual error between the public and private subspace, we create an additional noise vector z' (Line 12), add it to the best public gradient (indexed with $\hat{j}$), and perform another comparison between private $f(x-\eta g_{\hat{j}})$ and private $f(x-\eta(g_{\hat{j}}+z'))$ (Line 14). While PAZO-S is motivated by the arguments under a convex f (Eq. (3)), we apply it to all the tasks and models that are non-convex.

3.4. Privacy guarantees of PAZO

The privacy guarantees of all three methods can be analyzed in the same way. At each iteration, we guarantee the L_2 sensitivity of the sum of the function queries by C, and we add Gaussian noise with variance $qC^2\sigma^2$ where q is the number of queries on the sampled data. Therefore, the privacy bound per iteration is the same for any q, following the n-fold composition corollary of the Gaussian mechanism (Dong et al., 2022). Applying standard Renyi DP accounting (Mironov, 2017) to compose across T rounds with sampling ratio b/n, we have that there exist constants c_1 and c_2 such that for any $\varepsilon<c_1 b^2 T/n^2$, all three Algorithms 1-3 are (ε,δ)-differentially private for any $\delta>0$ if $\sigma\geq c_2\frac{b\sqrt{T\log(1/\delta)}}{n\varepsilon}$.

4. Convergence Analysis

In this section, we study the convergence properties of three PAZO algorithms. We first define the similarity between public and private data through the distance between the full gradients as follows.

Definition 4.1 (γ-similarity)**.** *Denote $\nabla f'(x_t)$ and $\nabla f(x_t)$ as the gradient for model x_t at time step t under the full public and private data, respectively. We call public and private data γ-similar if $\|\nabla f'(x_t)-\nabla f(x_t)\|\leq\gamma$ for all t.*

Algorithm 3 PAZO-S

```
1: Input: Same as Algorithm 2, and perturbation scale $\epsilon$
2: for $t=0,\cdots,T-1$ do
3:     Sample a batch $B$ of private training data $\{\xi_1,\ldots,\xi_b\}$
4:     Sample $k$ batches of public data and obtain their gradients $\{g_1,\ldots,g_k\}$
5:     $\tilde{g}\leftarrow 0^d$
6:     for $j=1,\ldots,k$ do
7:         Sample $z\sim\frac{1}{b}\mathcal{N}(0,(k+1)C^2\sigma^2)$
8:         $f_j\leftarrow\frac{1}{b}\sum_{i=1}^{b}\text{clip}_C(f(x_t-\eta g_j;\xi_i))+z$
9:     end for
10:    $\hat{j}\leftarrow\text{argmin}_{j\in[k]}f_j$
11:    $g_{k+1}\leftarrow g_{\hat{j}}+z'$ where $z'\sim\mathcal{N}(0,\epsilon^2 I_d)$
12:    Sample $z\sim\frac{1}{b}\mathcal{N}(0,(k+1)C^2\sigma^2)$
13:    $f_{k+1}\leftarrow\frac{1}{b}\sum_{i=1}^{b}\text{clip}_C(f(x_t-\eta g_{k+1};\xi_i))+z$
14:    $j^*\leftarrow\text{argmin}_{j\in[k+1]}f_j$
15:    $x_{t+1}\leftarrow x_t-\eta g_{j^*}$
16: end for
```

We note that such similarity is defined on top of the full gradients, a weaker requirement than defining on the stochastic gradients. There are previous similarity metrics based on coordinate-wise gradient norm alignment (Li et al., 2022). Together with their assumption on the bounded gradient norm, their similarity condition implies ours and is thus a stronger condition than ours.

Assumption 1. *The objective f evaluated on all private training data satisfies $\|f(x)-f(y)\|\leq M\|x-y\|,\forall x,\, y\in\mathbb{R}^d$.*

Assumption 2. *$f(x;\xi)$ is L-smooth for any $x\in\mathbb{R}^d$ and any subset data ξ.*

Assumption 3. *The variance of private stochastic gradients is bounded, i.e., $\mathbb{E}[\|\nabla f(x_t;\xi_i)-\nabla f(x_t)\|^2]\leq\sigma_1^2$ for any private sample ξ_i and any t.*

Assumption 4. *The variance of public stochastic gradients is bounded, i.e., $\mathbb{E}[\|\nabla f'(x_t;\xi_i')-\nabla f'(x_t)\|^2]\leq\sigma_2^2$ for any public sample ξ_i' and any t.*

Theorem 4.1 (Convergence of PAZO-M)**.** *Assume public and private data are γ-similar. Let Assumptions 1-4 hold. For possibly non-convex $f(\cdot)$, running Algorithm 1 under a fixed learning rate for T rounds gives*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|^2]\leq O\left(\frac{1}{T}\right)+O\big((1-\alpha)\lambda+\alpha\gamma^2\big)$$
$$+O\big((1-\alpha)\lambda^2+\alpha(\gamma+\lambda+\lambda\gamma)\big)$$
$$+O\left((1-\alpha)^2\left(\frac{\sigma_1^2}{b}+\frac{\sigma^2}{b^2}\right)+\alpha^2\frac{\sigma_2^2}{b'}\right). \tag{4}$$

We present several discussions on the results. First, the error reduces when γ is small. When $\alpha=0$, PAZP-M reduces to vanilla DPZero (Zhang et al., 2023) and our upper bound matches that of the full-rank and stochastic version of DPZero. We further analyze in Appendix B.2 that there exists an optimal $\alpha\in(0,1)$ that results in the best upper bound when γ does not exceed a threshold. Second, here we assume fixed λ, while the terms involving λ on RHS would vanish if we use decaying λ. Third, there are terms related to the variance of the stochastic gradients, which is standard when a constant learning rate (Zaheer et al., 2018) is assumed and would reduce as batch-size b and b' increase.

Theorem 4.2 (Convergence of PAZO-P)**.** *Let assumptions in Theorem 4.1 hold. For possibly non-convex $f(\cdot)$, running Algorithm 2 under a fixed learning rate for T rounds gives*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|^2]$$
$$\leq O\left(\frac{1}{T}\right)+O\left(\sqrt{\frac{\sigma_2^2}{b}+\gamma^2}+\lambda+\lambda^2+\frac{\sigma_1^2}{b}+\frac{\sigma^2}{b^2}\right). \tag{5}$$

Similar to Theorem 4.1, choosing a decaying λ makes the terms related to $\lambda+\lambda^2$ vanish to zero. Other terms are due to γ-similarity and variance of stochastic gradients.

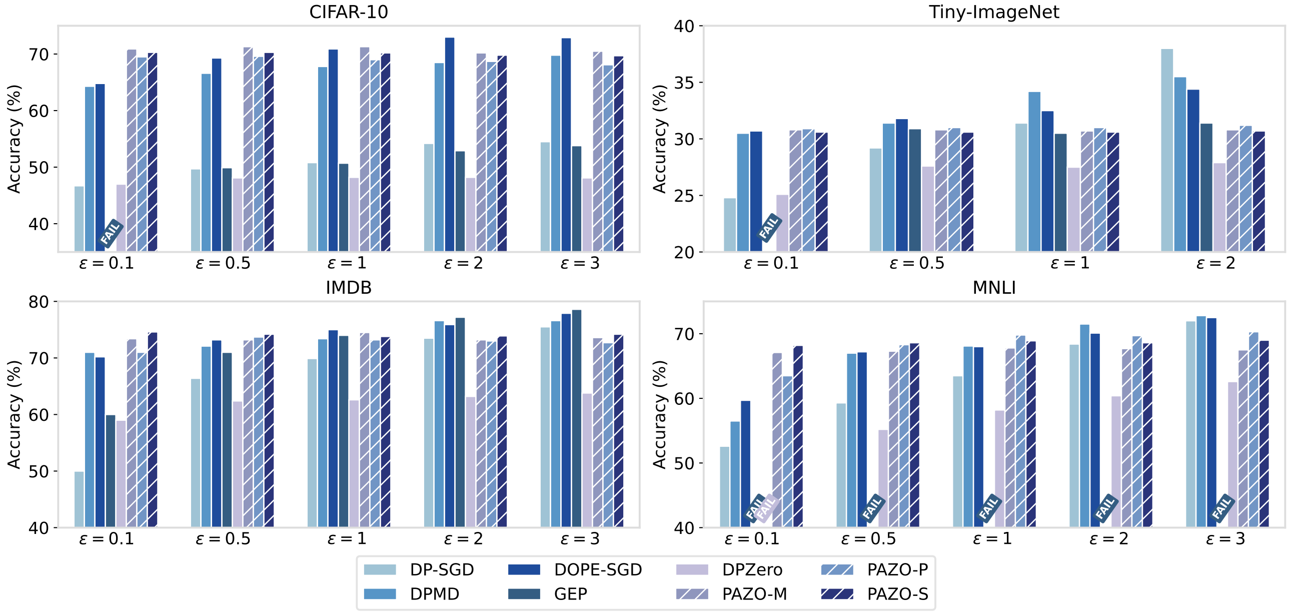


Figure 2: Performance of PAZO and the baselines on four settings. It shows that (1) all three PAZO variants outperform DPZero across all datasets, (2) all of the first-order methods (DP-SGD, DPMD, DOPE-SGD, and GEP), with or without public data, are more sensitive to smaller ε's than zeroth-order ones, and (3) when ε's are small, PAZO is superior to first-order baselines. "Fail" indicates failure to converge; the detailed accuracy numbers are in Tables 1-4 in the appendix.

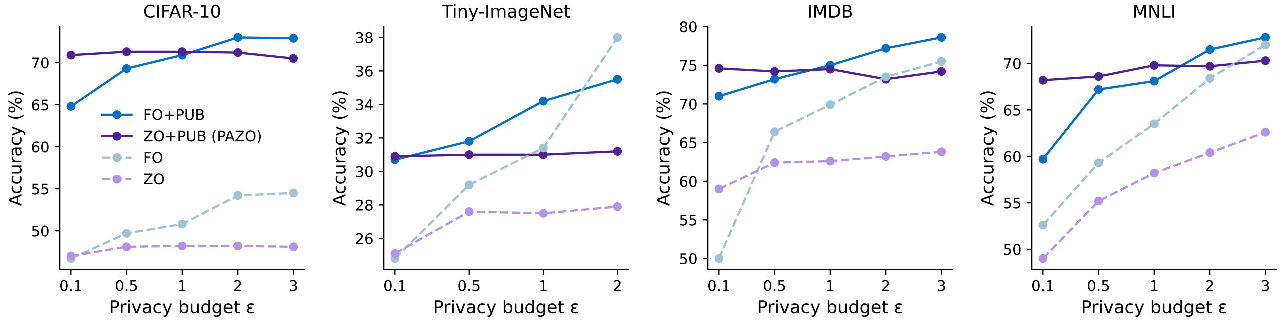


Figure 3: We compare the best private zeroth-order (ZO) methods with the best private first-order (FO) methods, with public data (+PUB) or without. Note that ZO+PUB is PAZO. It shows that (1) with or without public data, the performance gap between ZO and FO decreases as ε decreases, (2) using public data expands the range of ε's where ZO methods outperform FO ones, and (3) ZO+PUB (PAZO) achieves better privacy/utility tradeoff than FO+PUB when ε's are small.

Theorem 4.3 (Convergence of PAZO-S)**.** *Let assumptions in Theorem 4.1 hold. For possibly non-convex $f(\cdot)$, running Algorithm 3 under a fixed learning rate for T rounds gives*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|^2]\leq O\left(\frac{1}{T}\right)+O\left(\gamma+\gamma^2+\frac{\sigma_2}{\sqrt{b'}}+\frac{\sigma_2^2}{b'}\right). \tag{6}$$

Similarly, when γ approaches zero, the remaining term $\sigma_2/\sqrt{b'}+\sigma_2^2/b'$ is due to stochastic public data sampling. We give complete statements and proofs in Appendix B.

5. Empirical Evaluation

In this section, we present the empirical performance of PAZO-{M,P,S} on both the vision and language domains, across pre-training, fine-tuning, and prompt tuning tasks. In Section 5.1, we introduce experiment setups including datasets and models. In Section 5.2, we present the privacy/utility tradeoffs of PAZO, showing that PAZO performs comparably to public data augmented first-order methods over a number of tasks in moderate privacy regimes and outperforms them in highly private regimes. In Section 5.3, we highlight the time efficiency of PAZO. In Section 5.4, we present the sensitivity study of the hyperparameters, showing that PAZO is non-sensitive to introduced hyperparameters. Our code is publicly available at `github.com/xuchengong/pazo`.

5.1. Experimental Setups

The settings of our experiments cover and closely follow the experiments in the existing DP literature, including (1) Training NFResNet18 on CIFAR-10 (Krizhevsky et al.) from scratch, (2) fine-tuning Places365 pre-trained ViT-S on Tiny-ImageNet (mnmoustafa & Ali, 2017), (3) training LSTM on IMDB (Maas et al., 2011) from scratch, and (4) prompt-tuning RoBERTa-base on MNLI (Williams et al., 2017). We introduce distribution shifts between private and public data, such as class imbalance and context shifts. The details of public data generation and the distribution shifts between public and private data are in Appendix C.1.

5.2. Improved Privacy/Utility Tradeoffs

First, we compare PAZO with vanilla zeroth-order methods and various strong first-order baselines with public data under various privacy budgets $\varepsilon=\{0.1,0.5,1,2,3\}$. In Figure 2, we compare with 1) DP-SGD, the plain first-order method without public data, 2) DPZero, the plain zeroth-order method without public data, and 3) the state-of-the-art first-order algorithms with public data, including DPMD (Amid et al., 2022), GEP (Yu et al., 2021), and DOPE-SGD (Nasr et al., 2023).

We observe that all three PAZO variants outperform DPZero across the four datasets, though there is not a single PAZO algorithm that dominates other PAZO instances in all settings. In addition, all of the first-order methods (DP-SGD, DPMD, DOPE-SGD, and GEP), with or without public data, are much more sensitive to more strict privacy requirements (smaller ε's) than zeroth-order ones. This suggests that PAZO (and zeroth-order methods in general) possess more robust privacy/utility tradeoffs than the first-order methods across model types, training types, and task domains. Under small ε', PAZO is superior to first-order baselines by a large margin. We provide concrete accuracy numbers in Tables $1-4$ in the appendix.

Furthermore, we report the performance of the best PAZO algorithm among three variants (denoted as 'ZO+PUB') and the performance of the best public data augmented first-order method (denoted as 'FO+PUB') under different ε's in Figure 3. The results indicate that although zeroth-order (ZO) may underperform first-order (FO) variants, if we augment both with public data, PAZO performs comparably or is superior to the best first-order approach with public data (FO+PUB), while being more efficient.

5.3. Time Efficiency

In this section, we present the time efficiency of PAZO. It is faster than first-order methods (with or without public data) due to exemption from per-sample gradient computation, and it converges faster than zeroth-order baselines.

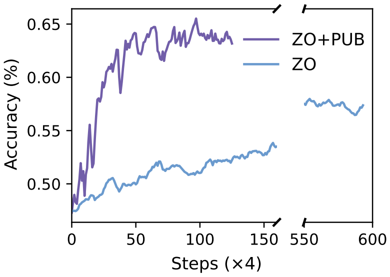


Figure 4: PAZO converges faster than DPZero on MNLI. The reported results are the smoothed test accuracy of PAZO-S with privacy budget ε=1.

#Iterations to converge. MeZO and DPZero present results with zeroth-order methods running 100× and 10× more steps than the first-order methods, but PAZO does not because of the assistance from public data. Figure 4 illustrates that public information significantly accelerates the convergence of zeroth-order methods. This is particularly favorable to differentially private training since smaller noise will be added due to smaller privacy consumption.

Runtime per iteration. Theoretically, we compare the number of different operations in each method in Table 6. Since the number of forward and backward passes in first-order methods depends on the size of the private batch-size, first-order methods can be dramatically slow since large-batch training is favorable in DP (McMahan et al., 2017; Yu et al., 2023). Empirically, we compare the speed of each method in terms of training time per iteration. Each experiment is conducted on one 48GB L40S GPU. For fair comparison, we maximally leverage the methods to speed up first-order DP, especially vectorization, just-in-time compilation, and static graph optimization (Subramani et al., 2021). In practice, due to the memory burden of parallelization and compilation overhead, a hybrid of `vmap` and sequential processing is often faster. We choose the fastest implementation for each first- and zeroth-order method under memory constraints. By comparing the utility/speed tradeoff (Figure 5), we observe that PAZO is comparable to or more performant than the baselines, while being 2∼16× faster in each training iteration.

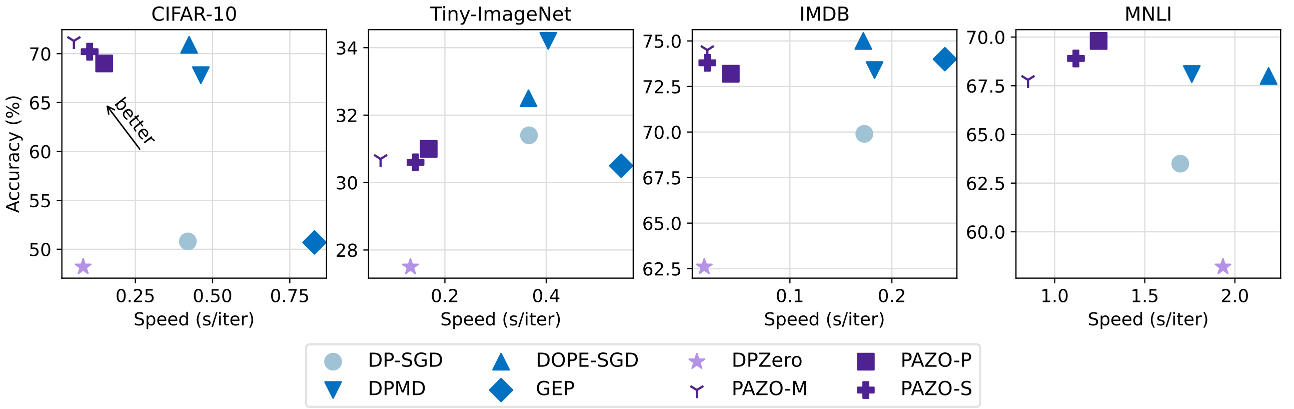
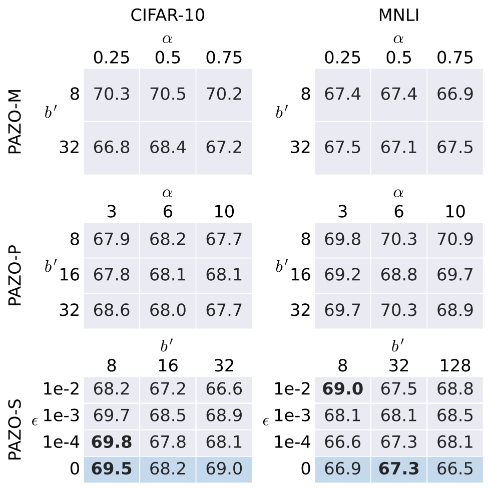


Figure 5: The utility/speed tradeoffs of different methods. It shows that PAZO is up to 16× faster in each training iteration than FO and FO+PUB while being comparably performant. The reported results are under privacy budget ε=1, and the detailed numbers are in Table 5.

5.4. Robustness to Hyperparameters



PAZO-M	CIFAR-10 α = 0.25	0.5	0.75	MNLI α = 0.25	0.5	0.75
b' = 8	70.3	70.5	70.2	67.4	67.4	66.9
32	66.8	68.4	67.2	67.5	67.1	67.5

PAZO-P	CIFAR-10 α = 3	6	10	MNLI α = 3	6	10
b' = 8	67.9	68.2	67.7	69.8	70.3	70.9
16	67.8	68.1	68.1	69.2	68.8	69.7
32	68.6	68.0	67.7	69.7	70.3	68.9

PAZO-S	CIFAR-10 b' = 8	16	32	MNLI b' = 8	32	128
ϵ = 1e-2	68.2	67.2	66.6	**69.0**	67.5	68.8
1e-3	69.7	68.5	68.9	68.1	68.1	68.5
1e-4	**69.8**	67.8	68.1	66.6	67.3	68.1
0	**69.5**	68.2	69.0	66.9	**67.3**	66.5

Figure 6: All PAZO methods are robust to different values of their introduced hyperparameters. Each number represents the best accuracy after the standard hyperparameters for zeroth-order private optimization (C and η) are tuned. Blue cells indicate PAZO-S performance without a noisy candidate.

Each method has its hyperparameters tuned using grid search, whose grid values are in Appendix C.3. Zeroth-order methods can sample q directions to reduce variance in each iteration, so we perform preliminary studies on $q \in \{1,5\}$ for each setting and choose $q=1$ if the performance gap is negligible. As presented in Appendix C.3, PAZO reduces the reliance on increased q compared to DPZero due to the guidance from public information.

Additionally, compared to the vanilla zeroth-order methods, PAZO has additional hyperparameters due to public data sampling, including the public batch-size b' and potentially the mixing coefficient α, number of candidates k, and perturbation scale ϵ. However, as in Figure 6 and Figure 8, we show that the performance of PAZO-{M,P,S} is robust to the values of these hyperparameters. In fact, a wide range of combinations of the introduced hyperparameter values can yield performance close to the best performance we report.

6. Conclusion and Future Work

We propose PAZO, a suite of public-data-assisted zeroth-order optimization methods for differentially private training. By leveraging modest amounts of public data and their gradients to guide zeroth-order updates, PAZO significantly improves the privacy/utility tradeoff over prior zeroth-order approaches while preserving their computational efficiencies. Through theoretical analysis and experiments across vision and language tasks, we demonstrate that PAZO closes the gap between zeroth- and first-order methods in moderate privacy regimes and even surpasses the best first-order baselines with public data under high privacy constraints. Our results position public-data-assisted zeroth-order optimization as a practical and scalable alternative for private training, especially in settings where private first-order methods are costly or infeasible. Future work could include sharpening the current convergence bounds by considering other similarity metrics and exploring a broader set of public and private dataset pairs in practical DP training applications.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Amid, E., Ganesh, A., Mathews, R., Ramaswamy, S., Song, S., Steinke, T., Suriyakumar, V. M., Thakkar, O., and Thakurta, A. Public data-assisted mirror descent for private model training. In *International Conference on Machine Learning*, pp. 517–535. PMLR, 2022.

Asi, H., Duchi, J., Fallah, A., Javidbakht, O., and Talwar, K. Private adaptive gradient methods for convex optimization. In *International Conference on Machine Learning*, pp. 383–392. PMLR, 2021.

Beltran, S. R., Tobaben, M., Jälkö, J., Loppi, N. A., and Honkela, A. Towards efficient and scalable implementation of differentially private deep learning.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

Brock, A., De, S., Smith, S. L., and Simonyan, K. High-performance large-scale image recognition without normalization. In *International conference on machine learning*, pp. 1059–1071. PMLR, 2021.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Chen, A., Zhang, Y., Jia, J., Diffenderfer, J., Liu, J., Parasyris, K., Zhang, Y., Zhang, Z., Kailkhura, B., and Liu, S. Deepzero: Scaling up zeroth-order optimization for deep model training. *arXiv preprint arXiv:2310.02025*, 2023.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 15–26, 2017.

Croce, F., Rebuffi, S.-A., Shelhamer, E., and Gowal, S. Seasoning model soups for robustness to adversarial and natural distribution shifts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12313–12323, 2023.

Dong, J., Roth, A., and Su, W. J. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(1):3–37, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, 2006.

Gu, B., Liu, G., Zhang, Y., Geng, X., and Huang, H. Optimizing large-scale hyperparameters via automated learning algorithm. *arXiv preprint arXiv:2102.09026*, 2021.

Hoffman, S. C., Chenthamarakshan, V., Wadhawan, K., Chen, P.-Y., and Das, P. Optimizing molecules using efficient queries from property evaluations. *Nature Machine Intelligence*, 4(1):21–31, 2022.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pp. 2137–2146. PMLR, 2018.

Kiefer, J. and Wolfowitz, J. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pp. 462–466, 1952.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Kurakin, A., Song, S., Chien, S., Geambasu, R., Terzis, A., and Thakurta, A. Toward training at imagenet scale with differential privacy, 2022. *URL https://arxiv.org/abs/2201.12328*, 2022.

Li, T., Zaheer, M., Reddi, S., and Smith, V. Private adaptive optimization with side information. In *International Conference on Machine Learning*, pp. 13086–13105. PMLR, 2022.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

Li, Z., Zhang, X., Zhong, P., Deng, Y., Razaviyayn, M., and Mirrokni, V. Addax: Utilizing zeroth-order gradients to improve memory efficiency and performance of sgd for fine-tuning language models. *arXiv preprint arXiv:2410.06441*, 2024.

Liu, Z., Lou, J., Bao, W., Hu, Y., Li, B., Qin, Z., and Ren, K. Differentially private zeroth-order methods for scalable large language model finetuning. *arXiv preprint arXiv:2402.07818*, 2024.

Ma, S. and Huang, H. Revisiting zeroth-order optimization: Minimum-variance two-point estimators and directionally aligned perturbations. In *The Thirteenth International Conference on Learning Representations*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Maheswaranathan, N., Metz, L., Tucker, G., Choi, D., and Sohl-Dickstein, J. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In *International Conference on Machine Learning*, pp. 4264–4273. PMLR, 2019.

Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J. D., Chen, D., and Arora, S. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

Meier, F., Mujika, A., Gauy, M. M., and Steger, A. Improving gradient estimation in evolutionary strategies with past descent directions. *arXiv preprint arXiv:1910.05268*, 2019.

Mironov, I. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

mnmoustafa and Ali, M. Tiny imagenet. https://kaggle.com/competitions/tiny-imagenet, 2017. Kaggle.

Nasr, M., Mahloujifar, S., Tang, X., Mittal, P., and Houmansadr, A. Effectively using public data in privacy preserving machine learning. In *International Conference on Machine Learning*, pp. 25718–25732. PMLR, 2023.

Sander, T., Yu, Y., Sanjabi, M., Durmus, A., Ma, Y., Chaudhuri, K., and Guo, C. Differentially private representation learning via image captioning. *arXiv preprint arXiv:2403.02506*, 2024.

Subramani, P., Vadivelu, N., and Kamath, G. Enabling fast differentially private sgd via just-in-time compilation and vectorization. *Advances in Neural Information Processing Systems*, 34:26409–26421, 2021.

Tang, X., Panda, A., Nasr, M., Mahloujifar, S., and Mittal, P. Private fine-tuning of large language models with zeroth-order optimization. *arXiv preprint arXiv:2401.04343*, 2024.

Verma, A., Subramanyam, A., Bangar, S., Lal, N., Shah, R. R., and Satoh, S. Certified zeroth-order black-box defense with robust unet denoiser. *arXiv preprint arXiv:2304.06430*, 2023.

Williams, A., Nangia, N., and Bowman, S. R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Yu, D., Zhang, H., Chen, W., and Liu, T.-Y. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. *arXiv preprint arXiv:2102.12677*, 2021.

Yu, Y., Sanjabi, M., Ma, Y., Chaudhuri, K., and Guo, C. Vip: A differentially private foundation model for computer vision. *arXiv preprint arXiv:2306.08842*, 2023.

Zaheer, M., Reddi, S., Sachan, D., Kale, S., and Kumar, S. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

Zhang, L., Li, B., Thekumparampil, K. K., Oh, S., and He, N. Dpzero: Private fine-tuning of language models without backpropagation. *arXiv preprint arXiv:2310.09639*, 2023.

Zhang, Q., Tran, H., and Cutkosky, A. Private zeroth-order nonsmooth nonconvex optimization. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=IzqZbNMZ0M.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhang, Y., Li, P., Hong, J., Li, J., Zhang, Y., Zheng, W., Chen, P.-Y., Lee, J. D., Yin, W., Hong, M., et al. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark. *arXiv preprint arXiv:2402.11592*, 2024b.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Zhou, Y., Wu, Z. S., and Banerjee, A. Bypassing the ambient dimension: Private sgd with gradient subspace identification. *arXiv preprint arXiv:2007.03813*, 2020.

A. Algorithm Details

A.1. PAZO-M norm alignment

To jusify why we sample the perturbation u from the sphere with radius $d^{\frac{1}{4}}$, we present the following analysis. For a random direction sampled uniformly from a sphere of radius r, the two-point estimator $g_\lambda(x)$ has the squared norm

$$\|g_\lambda(x)\|^2 = \left(\frac{f(x+\lambda u)-f(x-\lambda u)}{2\lambda}\right)^2 r^2.$$

Using a Taylor expansion of f and ignoring $O(\lambda^2)$ terms, we have $f(x\pm\lambda u)\approx f(x)\pm\lambda\nabla f(x)^\top u$ and thus

$$\|g_\lambda(x)\|^2 \approx (\nabla f(x)^\top u)^2 r^2.$$

Since $\mathbb{E}_u[uu^\top]=\frac{r^2}{d}I_d$,

$$\mathbb{E}_u[\|g_\lambda(x)\|^2]\approx r^2\mathbb{E}_u[(\nabla f(x)^\top u)^2]=r^2\nabla f(x)^\top\mathbb{E}_u[uu^\top]\nabla f(x)=\frac{r^4}{d}\|\nabla f(x)\|^2.$$

We thus have $\mathbb{E}_u[\|g_\lambda(x)\|^2]\approx\|\nabla f(x)\|^2$ if $r=d^{\frac{1}{4}}$.

A.2. PAZO-P perturbation sampling

We visualize the sampled perturbation set of the vanilla zeroth-order methods and PAZO-P as follows. We set $d=3,k=2$ and generate $G\in\mathbb{R}^{3\times 2}$ with normalized columns to represent the public gradients. The vanilla zeroth-order method samples the perturbations u in the full-dimensional sphere ($\mathbb{R}^3$), while PAZO-P samples in the column space of G. When G is orthonormal, we sample fairly in every direction in the public gradient subspace; when G is not orthonormal, we have larger effective learning rates in the directions in which the public gradients agree.

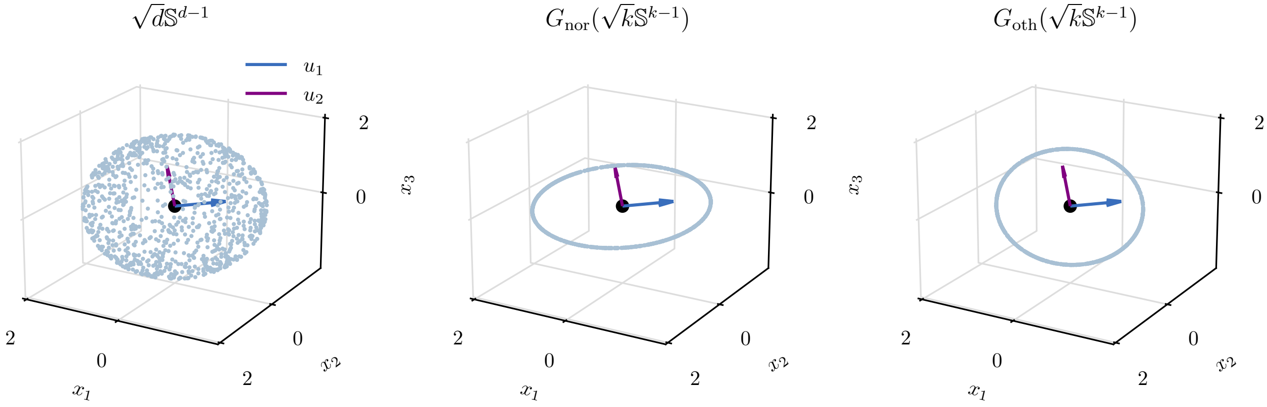


Figure 7: Comparison of the sampled perturbations in full-dimensional space and the public gradient subspace. u_1 and u_2 denote the top-2 left singular vectors of normalized G. *Left*: Vanilla zeroth-order perturbation sampling from $\sqrt{d}\mathbb{S}^{d-1}$. *Middle*: Sampling from $G(\sqrt{k}\mathbb{S}^{k-1})$ where G has normalized columns, which is functionally the border of a sphere elongated in the directions of top-k public gradient singular vectors. *Right*: Sampling from $G(\sqrt{k}\mathbb{S}^{k-1})$ where G is orthonormalized.

B. Detailed Convergence Analysis

B.1. Lemmas

Lemma B.1. *Let the private and public data be γ-similar and Assumption 3 and 4 hold. Denote $b:=|B|$ and $b':=|B'|$ as the private and public batch-size, respectively. Denote $g_t:=\nabla f(x_t)$ and $g'_t:=\nabla f'(x_t)$ as the gradient under full private and public data, respectively. Due to the stochasticity of sampling, the private and public batch gradients are*

$$\nabla f(x_t;B_t)=\frac{1}{b}\sum_{i\in B_t}(g_t+\zeta_{t,i}) \quad \textit{and} \quad \nabla f'(x_t;B_t')=\frac{1}{b'}\sum_{i\in B_t'}(g_t'+\zeta_{t,i}')$$

where $\zeta_{t,i}$ is independently sampled from some noise distribution $\mathcal{D}$ with zero mean and variance σ_1^2; $\zeta_{t,i}'$ is independently sampled from some noise distribution $\mathcal{D}'$ with zero mean and variance σ_2^2; B_t and B_t' are private and public batch at step t, respectively; $|B|$ and $|B'|$ are private and public batch-size, respectively. So we have

$$\begin{aligned}\mathbb{E}[\|\nabla f(x_t;B_t)-\nabla f'(x_t;B_t')\|]^2 &\leq \mathbb{E}[\|\nabla f(x_t;B_t)-\nabla f'(x_t;B_t')\|^2]\\ &=\mathbb{E}[\|g_t-g_t'\|^2]+\mathbb{E}\left[\left\|\frac{1}{b}\sum_{i\in B_t}\zeta_{t,i}\right\|^2\right]+\mathbb{E}\left[\left\|\frac{1}{b'}\sum_{i\in B_t'}\zeta_{t,i}'\right\|^2\right]\\ &\leq \gamma^2+\frac{\sigma_1^2}{b}+\frac{\sigma_2^2}{b'}\end{aligned}$$

where the first inequality is due to Jensen's inequality. Additionally,

$$\mathbb{E}[\|\nabla f(x_t;B_t)\|^2]=\mathbb{E}\left[\left\|g_t+\frac{1}{b}\sum_{i\in B_t}\zeta_{t,i}\right\|^2\right]=\mathbb{E}[\|g_t\|^2]+\mathbb{E}\left[\left\|\frac{1}{b}\sum_{i\in B_t}\zeta_{t,i}\right\|^2\right]\leq M^2+\frac{\sigma_1^2}{b}$$

Lemma B.2 (Zhang et al. (2023), Lemma C.1 and 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 $\sqrt{d}\mathbb{B}^d=\{x\in\mathbb{R}^d|\|x\|\leq\sqrt{d}\}$. Let $a\in\mathbb{R}^d$ be some fixed vector independent of u. We have*

1. $\mathbb{E}_u[u]=0$ *and* $\mathbb{E}_u[uu^\top]=I_d$.
2. $\mathbb{E}_u[u^\top a]=0$, $\mathbb{E}_u[(u^\top a)^2]=\|a\|^2$, *and* $\mathbb{E}_u[(u^\top a)u]=a$.
3. *For any function $f(x)$:$\mathbb{R}^d\rightarrow\mathbb{R}$ and $\lambda>0$, we define its zeroth-order gradient estimator as $g_\lambda(x)=\frac{f(x+\lambda u)-f(x-\lambda u)}{2\lambda}u$ and the smoothed function as $f_\lambda(x)=\mathbb{E}_u[f(x+\lambda u)]$. Then the following properties hold*
   - *(a) $f_\lambda(x)$ is differentiable and $\mathbb{E}_u[g_\lambda(x)]=\nabla f_\lambda(x)$.*
   - *(b) If $f(x)$ is L-smooth, then we have*

$$\|\nabla f(x)-\nabla f_\lambda(x)\|\leq\frac{L}{2}\lambda d^{3/2},$$

$$\mathbb{E}_u[\|g_\lambda(x)\|^2]\leq 2d\cdot\|\nabla f(x)\|^2+\frac{L^2}{2}\lambda^2 d^3.$$

B.2. Convergence of PAZO-M

Theorem B.3 (Full statement of Theorem 4.1)**.** *Let the private and public data be γ-similar and Assumption 1, 2, 3, and 4 hold. For possibly non-convex f, running Algorithm 1 for T rounds using a fixed step size $\eta=\frac{1}{4L(1-\alpha)(d-d\alpha+\alpha)}$ gives*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|^2]\leq&\frac{16L(1-\alpha)(d-d\alpha+\alpha)(f(x_0)-f(x_*))}{T}+\frac{(1-\alpha)L^2\lambda^2d^3}{2}+\frac{L^2\lambda^2d^2}{4}\\ &+\frac{\sigma_1^2}{b}+\frac{\sigma^2C^2}{2b^2}+\frac{\alpha[\gamma L\lambda d^{\frac{3}{2}}/2+(\gamma+L\lambda d^{\frac{3}{2}}/2)M]}{(1-\alpha)d}+2\alpha\gamma^2+\frac{\alpha^2\sigma_2^2}{2(1-\alpha)^2b'd}.\end{aligned}$$

Proof. Assume that clipping does not happen, then the update rule is $x_{t+1}-x_t=-\eta_t((1-\alpha)(\Delta(x_t;u_t;B_t)+z_t/b)u_t+\alpha g'(x_t;B_t'))$ where

$$\Delta(x_t;u_t;B_t)=\frac{1}{b}\sum_{\xi_i\in B_t}\frac{f(x_t+\lambda u_t;\xi_i)-f(x_t-\lambda u_t;\xi_i)}{2\lambda}.$$

At a step t, let x_t be a fixed parameter. We apply the update to the property of L-smooth objectives and take expectation over all the randomness at this iteration, i.e., $\mathbb{E}_t{:=}\mathbb{E}_{u_t,z_t,B_t,B'_t}$. We have

$$\begin{aligned}
&\mathbb{E}_t[f(x_{t+1})]\\
\le& f(x_t)+\langle\nabla f(x_t),\mathbb{E}_t[x_{t+1}-x_t]\rangle+\frac{L}{2}\mathbb{E}_t[\|x_{t+1}-x_t\|^2]\\
=& f(x_t)-(1-\alpha)\eta_t\nabla\underbrace{f(x_t)^\top\mathbb{E}_t[\Delta(x_t;u_t,B_t)u_t]}_{T_1}+\frac{(1-\alpha)^2L\eta_t^2\sqrt{d}}{2}\underbrace{\mathbb{E}_t[\Delta(x_t;u_t,B_t)^2]}_{T_2}\\
&\underbrace{+\frac{\alpha^2L\eta_t^2}{2}\mathbb{E}_t[\|g'(x_t;B'_t)\|^2]-\alpha\eta_t\nabla f(x_t)^\top g'_t+\alpha(1-\alpha)L\eta_t^2\mathbb{E}_t\left[\Delta(x_t;u_t,B_t)u_t^\top g'(x_t;B'_t)\right]}_{T_3}\\
&+\frac{(1-\alpha)^2L\eta_t^2d\sigma^2C^2}{2b^2}
\end{aligned}$$

For T_1, note that $\mathbb{E}_t[\Delta(x_t;u_t,B_t)u_t]{=}\mathbb{E}_t[u_tu_t^\top\nabla f(x_t)]{=}\frac{1}{\sqrt{d}}\nabla f(x_t)$ for $u_t{\sim}d^{\frac{1}{4}}\mathbb{S}^{d-1}$. We thus apply Lemma B.2 $(iii)(b)$ to obtain

$$\begin{aligned}
-\nabla f(x_t)^\top\mathbb{E}_t[\Delta(x_t;u_t,B_t)u_t]=&-\nabla f(x_t)^\top\mathbb{E}_{u_t}[\Delta(x_t;u_t)u_t]\\
=&-\langle\nabla f(x_t)^\top,\nabla f(x_t)+\mathbb{E}_{u_t}[\Delta(x_t;u_t)u_t]-\nabla f(x_t)\rangle\\
\le&-\|\nabla f(x_t)\|^2+\|\nabla f(x_t)\|\|\mathbb{E}_{u_t}[\Delta(x_t;u_t)u_t]-\nabla f(x_t)\|\\
\le&-\|\nabla f(x_t)\|^2+\|\nabla f(x_t)\|\left[\underbrace{\left\|\mathbb{E}_{u_t}[\Delta(x_t;u_t)u_t]-\frac{1}{\sqrt{d}}\nabla f(x_t)\right\|}_{T_5}+\left(1-\frac{1}{\sqrt{d}}\right)\|\nabla f(x_t)\|\right]
\end{aligned}\tag{7}$$

where T_5 has

$$\begin{aligned}
\left\|\frac{1}{\sqrt{d}}\nabla f(x_t)-\mathbb{E}_{u_t}[\Delta(x_t;u_t)u_t]\right\|\le&\mathbb{E}_t\left[\left\|\left(\nabla f(x_t)^\top u_t-\frac{f(x_t+\lambda u_t)-f(x_t-\lambda u_t)}{2\lambda}\right)u_t\right\|\right]\\
=&\frac{d^{\frac{1}{4}}}{2\lambda}\mathbb{E}_t\left[|\left(f(x_t+\lambda u_t)-f(x_t-\lambda u_t)-2\lambda\nabla f(x_t)^\top u_t\right)|\right]\\
\le&\frac{d^{\frac{1}{4}}}{2\lambda}\mathbb{E}_t\left[|\left(f(x_t+\lambda u_t)-f(x_t)-\lambda\nabla f(x_t)^\top u_t\right)|\right]\\
&+\frac{d^{\frac{1}{4}}}{2\lambda}\mathbb{E}_t\left[|\left(f(x_t)-f(x_t-\lambda u_t)-\lambda\nabla f(x_t)^\top u_t\right)|\right]\\
\le&\frac{L\lambda d^{\frac{3}{4}}}{2}
\end{aligned}$$

due to L-smoothness applied to the last inequality. Therefore, $-\nabla f(x_t)^\top\mathbb{E}_t[\Delta(x_t;u_t,B_t)u_t]{\le}-\frac{1}{\sqrt{d}}\|\nabla f(x_t)\|+\frac{L\lambda d^{\frac{3}{4}}}{2}M$.

For T_2, note that per-sample L-smoothness implies batch L-smoothness. Therefore, we follow Zhang et al. (2023) by noting that

$$\begin{aligned}\Delta(x_t;u_t,B_t)^2=&\frac{(f(x_t+\lambda u_t;B_t)-f(x_t-\lambda u_t;B_t)-2\lambda u_t^\top\nabla f(x_t;B_t)+2\lambda u_t^\top\nabla f(x_t;B_t))^2}{4\lambda^2}\\ \overset{(a)}{\le}&\frac{(f(x_t+\lambda u_t;B_t)-f(x_t-\lambda u_t;B_t)-2\lambda u_t^\top\nabla f(x_t;B_t))^2+(2\lambda u_t^\top\nabla f(x_t;B_t))^2}{2\lambda^2}\\ \overset{(b)}{\le}&\frac{(f(x_t+\lambda u_t;B_t)-f(x_t;B_t)-\lambda u_t^\top\nabla f(x_t;B_t))^2}{\lambda^2}\\ &+\frac{(f(x_t;B_t)-f(x_t-\lambda u_t;B_t)-\lambda u_t^\top\nabla f(x_t;B_t))^2}{\lambda^2}+2(u_t^\top\nabla f(x_t;B_t))^2\\ \overset{(c)}{\le}&\frac{L^2\lambda^2 d}{2}+2(u_t^\top\nabla f(x_t;B_t))^2\end{aligned}$$

where (a) and (b) are implied by $(a+b)^2\le 2(a^2+b^2)$ and (c) uses the facts $|f(x+\lambda u)-f(x)-\lambda u^\top\nabla f(x)|\le L\lambda^2 d/2$ and $|f(x)-f(x-\lambda u)-\lambda u^\top\nabla f(x)|\le L\lambda^2 d/2$ due to L-smoothness. Therefore,

$$\begin{aligned}\mathbb{E}_{u_t}[\Delta(x_t;u_t,B_t)^2]\overset{(a)}{=}&\frac{L^2\lambda^2 d}{2}+\frac{2}{\sqrt{d}}\|\nabla f(x_t;B_t)\|^2\\ \le&\frac{L^2\lambda^2 d}{2}+\frac{2}{\sqrt{d}}\|\nabla f(x_t)\|^2+\frac{2\sigma_1^2}{b\sqrt{d}}\end{aligned}\tag{8}$$

where (a) follows Lemma B.2 (ii).

For T_3, applying the equalities

$$\mathbb{E}_{B_t'}[\|g'(x_t;B_t')\|^2]=\|g'\|^2+\frac{\sigma_2^2}{b'},$$

$$\nabla f(x_t)^\top g_t'=\frac{1}{2}(\|g_t'\|^2+\|\nabla f(x_t)\|^2-\|g_t'-\nabla f(x_t)\|^2),$$

$$\begin{aligned}\mathbb{E}_{u_t,B_t,B_t'}[\Delta(x_t;u_t,B_t)u_t^\top g'(x_t;B_t')]=&\nabla f_\lambda(x_t)^\top g_t'\\ =&\frac{1}{2}(\|g_t'\|^2+\|\nabla f_\lambda(x_t)\|^2-\|g_t'-\nabla f_\lambda(x_t)\|^2)\end{aligned}$$

gives us

$$T_3=\frac{\alpha L\eta_t^2}{2}\left[\left(1-\frac{1}{L\eta_t}\right)\|g_t'\|^2+(1-\alpha)\|\nabla f_\lambda(x_t)\|^2-(1-\alpha)\|g_t'-\nabla f_\lambda(x_t)\|^2\right]+T_4,\tag{9}$$

where

$$\begin{aligned}T_4=&\frac{\alpha\eta_t}{2}\|g_t'-\nabla f(x_t)\|^2+\frac{\alpha^2 L\eta_t^2\sigma_2^2}{2b'}-\frac{\alpha\eta_t}{2}\|\nabla f(x_t)\|^2\\ \le&\frac{\alpha\eta_t}{2}\gamma^2+\frac{\alpha^2 L\eta_t^2\sigma_2^2}{2b'}-\frac{\alpha\eta_t}{2}\|\nabla f(x_t)\|^2\end{aligned}\tag{10}$$

We take α and η_t so that $\alpha L\eta_t<1$, which implies $1-\frac{1}{L\eta_t}<1-\alpha$. We thus have

$$\begin{aligned}T_3\le&\frac{\alpha(1-\alpha)L\eta_t^2}{2}\left[\|g_t'\|^2+\|\nabla f_\lambda(x_t)\|^2-\|g_t'-\nabla f_\lambda(x_t)\|^2\right]+T_4\\ =&\alpha(1-\alpha)\langle g_t',\nabla f_\lambda(x_t)\rangle+T_4\\ \le&\alpha(1-\alpha)\|g_t'\|\|\nabla f_\lambda(x_t)\|+T_4\\ \le&\alpha(1-\alpha)(\|g_t'-\nabla f(x_t)\|+\|\nabla f(x_t)\|)(\|\nabla f_\lambda(x_t)-\nabla f(x_t)\|+\|\nabla f(x_t)\|)+T_4\\ \le&\alpha(1-\alpha)(\gamma L\lambda d^{\frac{3}{4}}/2+(\gamma/\sqrt{d}+L\lambda d^{\frac{3}{4}}/2)M+\|\nabla f(x_t)\|^2/\sqrt{d})+T_4\end{aligned}\tag{11}$$

Combining T_1 (7), T_2 (8), T_3 (11), and T_4 (10) yields

$$\begin{aligned}&\left[\frac{\eta_t(1-\alpha)}{\sqrt{d}}+\frac{\eta_t\alpha}{2}-L\eta_t^2(1-\alpha)^2-\frac{L\eta_t^2\alpha(1-\alpha)}{\sqrt{d}}\right]\|\nabla f(x_t)\|^2\\ &\leq f(x_t)-\mathbb{E}_t[f(x_{t+1})]+\frac{(1-\alpha)L\eta_t\lambda d^{\frac{3}{4}}M}{2}+\frac{(1-\alpha)^2L\eta_t^2\sigma_1^2}{b}\\ &\quad+\frac{(1-\alpha)^2L^3\eta_t^2\lambda^2d^{\frac{3}{2}}}{4}+\frac{(1-\alpha)^2L\eta_t^2\sigma^2C^2\sqrt{d}}{2b^2}+\frac{\alpha\eta_t\gamma^2}{2}\\ &\quad+\frac{\alpha^2L\eta_t^2\sigma_2^2}{2b'}+\frac{\alpha(1-\alpha)L^2\eta_t^2\gamma\lambda d^{\frac{3}{4}}}{2}+\alpha(1-\alpha)L\eta_t^2M\left(\frac{\gamma}{\sqrt{d}}+\frac{L\lambda d^{\frac{3}{4}}}{2}\right).\end{aligned}$$

Choosing $\eta_t=\frac{2(1-\alpha)+\alpha\sqrt{d}}{4L(1-\alpha)(4L((1-\alpha)^2\sqrt{d}+\alpha(1-\alpha))}$, we have $\alpha L\eta_t<1$ if $\alpha<1-\frac{3\sqrt{d}-3}{3\sqrt{d}-2}$. Denote $\mathbb{E}_{<t}:=\mathbb{E}_{u_{<t},z_{<t},B_{<t},B'_{<t}}$ where $u_{<t}$ is the set $\{u_0,...,u_{t-1}\}$ and similarly for $z_{<t}$, $B_{<t}$, and $B'_{<t}$. Then we have

$$\begin{aligned}\mathbb{E}_{<t}[\|\nabla f(x_t)\|^2]\leq&16L(1-\alpha)(d-d\alpha+\alpha)\mathbb{E}_{<t+1}[f(x_t)-f(x_{t+1})]+\frac{(1-\alpha)L^2\lambda^2d^3}{2}\\ &+\left(L^2\lambda^2d^3+\frac{4\sigma_1^2d}{b}\right)\frac{1-\alpha}{4(d-d\alpha+\alpha)}+\frac{(1-\alpha)d\sigma^2C^2}{2b^2(d-d\alpha+\alpha)}+\\ &\frac{\alpha[\gamma L\lambda d^{\frac{3}{2}}/2+(\gamma+L\lambda d^{\frac{3}{2}}/2)M]}{d-d\alpha+\alpha}+2\alpha\gamma^2+\frac{\alpha^2\sigma_2^2}{2b'(1-\alpha)(d-d\alpha+\alpha)}.\end{aligned}$$

We sum up from $t=0$ to $T-1$, and divide both sides by T to obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{<t}[\|\nabla f(x_t)\|^2]\leq&\frac{16\sqrt{d}L\mathbb{E}_{<t+1}[f(x_0)-f(x_*)]}{T}\frac{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)}{(2(1-\alpha)+\alpha\sqrt{d})^2}+2L\lambda d^{\frac{5}{4}}M\frac{1-\alpha}{2(1-\alpha)+\alpha\sqrt{d}}\\ &+2\sqrt{d}\gamma^2\frac{\alpha}{2(1-\alpha)+\alpha\sqrt{d}}+\left[\frac{L^2\lambda^2d^2}{4}+\frac{\sigma_1^2\sqrt{d}}{b}+\frac{d\sigma^2C^2}{2b^2}\right]\frac{1-\alpha}{(1-\alpha)\sqrt{d}+\alpha}\\ &+\frac{\sqrt{d}\sigma_2^2}{2b'}\frac{\alpha^2}{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)}+\left[\frac{L\lambda d^{\frac{5}{4}}\gamma}{2}+\left(\gamma+\frac{L\lambda d^{\frac{5}{4}}}{2}\right)M\right]\frac{\alpha}{(1-\alpha)\sqrt{d}+\alpha}\end{aligned}\tag{12}$$

□

To interpret this result, we discuss two scenarios of α. First, when the public data brings no helpful information to private training, we can set $\alpha=0$ to reduce our update rule to DPZero. If we made the same assumption as in Zhang et al. (2023), including per-sample Lipchitz, effective low-rankness, and full private data sampling, our Eq. (13) becomes

$$O\left(\frac{1}{T}\right)+O(\lambda^2)+O(\sigma^2),$$

which is the same utility bound as theirs.

Second, when public data are of good quality, we seek to find $\alpha\in[0,1-\frac{3\sqrt{d}-3}{3\sqrt{d}-2})$ that minimizes the RHS of Eq. (13). We analyze the existance of α in two scenarios: (1) Use constant λ (Zhang et al., 2023) and (2) use decaying $\lambda=O(1/t)$ (Duchi et al., 2015).

When λ is constant as in DPZero, we group all the terms dependent on α and denote

$$\begin{aligned}h(\alpha)=&A_1\frac{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)}{(2(1-\alpha)+\alpha\sqrt{d})^2}+A_2\frac{1-\alpha}{2(1-\alpha)+\alpha\sqrt{d}}+A_3\frac{\alpha}{2(1-\alpha)+\alpha\sqrt{d}}\\ &+A_4\frac{1-\alpha}{(1-\alpha)\sqrt{d}+\alpha}+A_5\frac{\alpha^2}{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)}+A_6\frac{\alpha}{(1-\alpha)\sqrt{d}+\alpha}\end{aligned}\tag{13}$$

where

$$A_1=\frac{16\sqrt{d}L\mathbb{E}_{<t+1}[f(x_0)-f(x_*)]}{T},A_2=2L\lambda d^{\frac{5}{4}}M,A_3=2\sqrt{d}\gamma^2$$

$$A_4=\frac{L^2\lambda^2d^2}{4}+\frac{\sigma_1^2\sqrt{d}}{b}+\frac{d\sigma^2C^2}{2b^2},A_5=\frac{\sqrt{d}\sigma_2^2}{2b'},\text{ and }A_6=\frac{L\lambda d^{\frac{5}{4}}\gamma}{2}+\left(\gamma+\frac{L\lambda d^{\frac{5}{4}}}{2}\right)M.$$

Denote $D_1=2(1-\alpha)+\alpha\sqrt{d}$ and $D_2=(1-\alpha)\sqrt{d}+\alpha$. Its derivative is

$$\begin{aligned}h'(\alpha)=&A_1\frac{D_1(-2\sqrt{d}+2\alpha\sqrt{d}-2\alpha+1)-2(1-\alpha)[(1-\alpha)\sqrt{d}+\alpha](\sqrt{d}-2)}{D_1^3}-A_2\frac{\sqrt{d}}{D_1^2}+A_3\frac{2}{D_1^2}\\&-A_4\frac{1}{D_2^2}+A_5\frac{\alpha(2-\alpha)D_2-\alpha^2(1-\alpha)(1-\sqrt{d})}{(1-\alpha)^2D_2^2}+A_6\frac{\sqrt{d}}{D_2^2}\end{aligned}$$

and

$$h'(0)=A_1\frac{1-d}{4}-A_2\frac{\sqrt{d}}{4}+A_3\frac{1}{2}-A_4\frac{1}{d}+A_6\frac{1}{\sqrt{d}}.$$

Since $h(\alpha)$ is not monotonously increasing, there exists $\alpha>0$ s.t. the error term is minimized and our bound is improved upon the vanilla zeroth-order method. For example, since h is smooth on $[0,1)$, we have

$$\begin{aligned}&h(0)\text{ decreases near }0\\\Leftrightarrow&h'(0)<0\\\Leftrightarrow&\sqrt{d}\gamma^2+\frac{L\lambda d^{\frac{5}{4}}+2M}{2\sqrt{d}}\gamma<\frac{4\sqrt{d}L(d-1)(f(x_0)-f(x_*))}{T}+(d^{\frac{7}{4}}-d^{\frac{3}{4}})\frac{L\lambda M}{2}+\frac{L^2\lambda^2d}{4}+\frac{\sigma_1^2}{b\sqrt{d}}+\frac{\sigma^2C^2}{2b^2}\end{aligned}\tag{14}$$

Note that the quadratic inequality (14) has solutions for $\gamma>0$ since the RHS of 14 is larger than 0. Denote its largest solution as γ_{max} and we have

$$\gamma<\gamma_{\text{max}}\Leftrightarrow h(0)\text{ decreases near }0\Leftrightarrow\exists\alpha\in\left(0,1-\frac{3\sqrt{d}-3}{3\sqrt{d}-2}\right)\text{ s.t. }h(\alpha)<h(0).$$

Alternatively, when λ decays, we group the non-vanishing terms related to α and obtain

$$\begin{aligned}h'(\alpha)=&A_1\frac{D_1(-2\sqrt{d}+2\alpha\sqrt{d}-2\alpha+1)-2(1-\alpha)[(1-\alpha)\sqrt{d}+\alpha](\sqrt{d}-2)}{D_1^3}+A_3\frac{2}{D_1^2}\\&-\left(\frac{\sigma_1^2\sqrt{d}}{b}+\frac{d\sigma^2C^2}{2b^2}\right)\frac{1}{D_2^2}+A_5\frac{\alpha(2-\alpha)D_2-\alpha^2(1-\alpha)(1-\sqrt{d})}{(1-\alpha)^2D_2^2}+\gamma M\frac{\sqrt{d}}{D_2^2}\end{aligned}$$

where

$$A_1=\frac{16\sqrt{d}L\mathbb{E}_{<t+1}[f(x_0)-f(x_*)]}{T},A_3=2\sqrt{d}\gamma^2,\text{ and }A_5=\frac{\sqrt{d}\sigma_2^2}{2b'}.$$

Similarly, we have

$$\begin{aligned}&h(0)\text{ decreases near }0\\\Leftrightarrow&h'(0)<0\\\Leftrightarrow&\sqrt{d}\gamma^2+\frac{M}{\sqrt{d}}\gamma<\frac{4\sqrt{d}L(d-1)(f(x_0)-f(x_*))}{T}+\frac{\sigma_1^2}{b\sqrt{d}}+\frac{\sigma^2C^2}{2b^2}\end{aligned}$$

which always has a largest solution $\gamma_{\text{max}}>0$. The remaining conclusion is the same as above.

In conclusion, PAZO-M offers the opportunity to leverage public gradients if they are close to the private gradients. In the worst case, we can recover the vanilla private zeroth-order utility guarantee by setting $\alpha=0$.

B.3. Convergence of PAZO-P

Theorem B.4 (Full statement of Theorem 4.2)**.** *Let the private and public data be γ-similar and Assumption 1, 2, 3, and 4 hold. For possibly non-convex f, running Algorithm 2 for T rounds using a fixed step size $\eta=\frac{1}{2Lk}$ gives*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|^2]\le\frac{4Lk}{T}\mathbb{E}[f(x_0)-f(x_*)]+2M\sqrt{2\left(\frac{\sigma_2^2}{b'}+\gamma^2\right)}+L\lambda k^{\frac{3}{2}}+\frac{L^2\lambda^2k^2}{4}+\frac{\sigma_1^2}{b}+\frac{\sigma^2C^2}{2b^2}.$$

Proof. Assume that clipping does not happen, then the update rule is $x_{t+1}-x_t=-\eta_t(\Delta(x_t;u_t;B_t)+z_t/b)G_tu_t$ where

$$\Delta(x_t;u_t;B_t)=\frac{1}{b}\sum_{\xi_i\in B_t}\frac{f(x_t+\lambda G_tu_t;\xi_i)-f(x_t-\lambda G_tu_t;\xi_i)}{2\lambda}.$$

At a step t, let x_t be a fixed parameter. We apply the update to the property of L-smooth objectives and take expectation over all the randomness at this iteration, i.e., $\mathbb{E}_t:=\mathbb{E}_{u_t,z_t,B_t,B_t'}$. We have

$$\begin{aligned}
&\mathbb{E}_t[f(x_{t+1})]\\
&\le f(x_t)+\langle\nabla f(x_t),\mathbb{E}_t[x_{t+1}-x_t]\rangle+\frac{L}{2}\mathbb{E}_t[\|x_{t+1}-x_t\|^2]\\
&=f(x_t)-\eta_t\langle\nabla f(x_t),\mathbb{E}_t[\Delta(x_t;u_t,B_t)G_tu_t]\rangle+\frac{L\eta_t^2}{2}\mathbb{E}_t[\|\Delta(x_t;u_t,B_t)G_tu_t\|^2]+\frac{L\eta_t^2}{2}\mathbb{E}_t\left[\left\|\frac{z_t}{b}G_tu_t\right\|^2\right]\\
&\overset{(a)}{=}f(x_t)-\eta_t\|\nabla f(x_t)\|^2+\eta_t\underbrace{\langle\nabla f(x_t),\nabla f(x_t)-\mathbb{E}_t[\Delta(x_t;u_t,B_t)G_tu_t]\rangle}_{T_1}\\
&\quad+\frac{L\eta_t^2k}{2}\underbrace{\mathbb{E}_t[\|\Delta(x_t;u_t,B_t)\|^2]}_{T_2}+\frac{L\eta_t^2\sigma^2C^2k}{2b^2},
\end{aligned}\tag{15}$$

where (a) is due to the orthonormality of G_t and $\|u_t\|=\sqrt{k}$.

For T_1, we proceed by

$$\begin{aligned}
&\langle\nabla f(x_t),\nabla f(x_t)-\mathbb{E}_t[\Delta(x_t;u_t,B_t)G_tu_t]\rangle\\
&\le\|\nabla f(x_t)\|\|\nabla f(x_t)-\mathbb{E}_t[\Delta(x_t;u_t,B_t)G_tu_t]\|\\
&\le\|\nabla f(x_t)\|[\underbrace{\left\|\nabla f(x_t)-\mathbb{E}_t[G_tG_t^\top\nabla f(x_t)]\right\|}_{T_3}+\underbrace{\left\|\mathbb{E}_t[G_tG_t^\top\nabla f(x_t)]-\mathbb{E}_t[\Delta(x_t;u_t,B_t)G_tu_t]\right\|}_{T_4}].
\end{aligned}$$

For a G_t, we denote its un-orthonormalized columns as $\{g'(x_t;B'_{t,1}),...,g'(x_t;B'_{t,k})\}$. Note that for any public candidate index $i\in[k]$, we have

(i) $g'(x_t;B'_{t,i})\in\mathrm{Col}(G_t)$

(ii) $$\begin{aligned}\mathbb{E}_t[\left\|g(x_t;B'_{t,i})-\nabla f(x_t)\right\|^2]&=\mathbb{E}_t[\left\|g(x_t;B'_{t,i})-g'_t+g'_t-\nabla f(x_t)\right\|^2]\\
&\overset{(a)}{\le}2\mathbb{E}_t[\left\|g(x_t;B'_{t,i})-g'_t\right\|^2]+\|g'_t-\nabla f(x_t)\|^2\\
&\overset{(b)}{\le}2(\sigma_2^2/b'+\gamma^2).\end{aligned}$$

where (a) holds due to $(a+b)^2\le2(a^2+b^2)$ and (b) follows the γ-similar assumption. Therefore,

$$\begin{aligned}\left(\mathbb{E}_t[\left\|\nabla f(x_t)-G_tG_t^\top\nabla f(x_t)\right\|]\right)^2&\overset{(a)}{\le}\mathbb{E}_t[\left\|\nabla f(x_t)-G_tG_t^\top\nabla f(x_t)\right\|^2]\\
&\overset{(b)}{\le}\mathbb{E}_t[\left\|\nabla f(x_t)-g(x_t;B'_{t,i})\right\|^2]\\
&\le2(\sigma_2^2/b'+\gamma^2),\end{aligned}$$

where (a) follows Jensen's inequality and (b) is due to the fact that $\left\|\nabla f(x_t) - G_t G_t^\top \nabla f(x_t)\right\| \le \|\nabla f(x_t) - x\|$ for any $x \in \mathrm{Col}(G_t)$.

For T_3, we thus have

$$\begin{aligned} \left\|\nabla f(x_t) - \mathbb{E}_t[G_t G_t^\top \nabla f(x_t)]\right\| &\le \mathbb{E}_t[\left\|\nabla f(x_t) - G_t G_t^\top \nabla f(x_t)]\right\| \\ &\le \sqrt{2(\sigma_2^2/b' + \gamma^2)}. \end{aligned} \tag{16}$$

For T_4, we have

$$\begin{aligned} &\left\|\mathbb{E}_t[G_t G_t^\top \nabla f(x_t)] - \mathbb{E}_t[\Delta(x_t;u_t,B_t) G_t u_t]\right\| \\ =&\mathbb{E}_t\left[\left\|\left(\nabla f(x_t)^\top G_t u_t - \frac{f(x_t + \lambda G_t u_t;B_t) - f(x_t - \lambda G_t u_t;B_t)}{2\lambda}\right) G_t u_t\right\|\right] \\ =&\frac{\sqrt{k}}{2\lambda}\mathbb{E}_t\left[|\big(f(x_t + \lambda G_t u_t;B_t) - f(x_t - \lambda G_t u_t;B_t) - 2\lambda \nabla f(x_t)^\top G_t u_t\big)|\right] \\ \le&\frac{\sqrt{k}}{2\lambda}\mathbb{E}_t\left[|\big(f(x_t + \lambda G_t u_t;B_t) - f(x_t;B_t) - \lambda \nabla f(x_t)^\top G_t u_t\big)|\right] \\ &+\frac{\sqrt{k}}{2\lambda}\mathbb{E}_t\left[|\big(f(x_t;B_t) - f(x_t - \lambda G_t u_t;B_t) - \lambda \nabla f(x_t)^\top G_t u_t\big)|\right] \\ \le&\frac{L\lambda k^{\frac{3}{2}}}{2} \end{aligned}$$

where the last inequality is due to L-smoothness. Therefore,

$$T_1 \le M\left(\sqrt{2(\sigma_2^2/b' + \gamma^2)} + \frac{L\lambda k^{\frac{3}{2}}}{2}\right) \tag{17}$$

For T_2, note that

$$\begin{aligned} &\Delta(x_t;u_t,B_t)^2 \\ =&\frac{(f(x_t + \lambda G_t u_t;B_t) - f(x_t - \lambda G_t u_t;B_t) - 2\lambda u_t^\top G_t^\top \nabla f(x_t;B_t) + 2\lambda u_t^\top G_t^\top \nabla f(x_t;B_t))^2}{4\lambda^2} \\ \overset{(a)}{\le}&\frac{(f(x_t + \lambda G_t u_t;B_t) - f(x_t - \lambda G_t u_t;B_t) - 2\lambda u_t^\top G_t^\top \nabla f(x_t;B_t))^2 + (2\lambda u_t^\top G_t^\top \nabla f(x_t;B_t))^2}{2\lambda^2} \\ \overset{(b)}{\le}&\frac{(f(x_t + \lambda G_t u_t;B_t) - f(x_t;B_t) - \lambda u_t^\top G_t^\top \nabla f(x_t;B_t))^2}{\lambda^2} \\ &+\frac{(f(x_t;B_t) - f(x_t - \lambda G_t u_t;B_t) - \lambda u_t^\top \nabla f(x_t;B_t))^2}{\lambda^2} + 2(u_t^\top G_t^\top \nabla f(x_t;B_t))^2 \\ \overset{(c)}{\le}&\frac{L^2\lambda^2 k^2}{2} + 2(u_t^\top G_t^\top \nabla f(x_t;B_t))^2, \end{aligned}$$

where (a) and (b) are implied by $(a+b)^2 \le 2(a^2+b^2)$ and (c) uses the facts $|f(x+\lambda u) - f(x) - \lambda u^\top \nabla f(x)| \le L\lambda^2 d/2$ and

$|f(x)-f(x-\lambda u)-\lambda u^\top \nabla f(x)|\le L\lambda^2 d/2$ due to L-smoothness. Therefore, applying Lemma B.2 (iii) gives us

$$\begin{aligned}
\mathbb{E}_t[\|\Delta(x_t;u_t,B_t)\|^2] &= \frac{L^2\lambda^2k^2}{2}+2\mathbb{E}_{B_t,B_t'}\mathbb{E}_{u_t}[(u_t^\top G_t^\top \nabla f(x_t;B_t))^2] \\
&= \frac{L^2\lambda^2k^2}{2}+2\mathbb{E}_{B_t,B_t'}[\left\|G_t^\top \nabla f(x_t;B_t)\right\|^2] \\
&= \frac{L^2\lambda^2k^2}{2}+2\mathbb{E}_{B_t,B_t'}[\nabla f(x_t;B_t)^\top G_t G_t^\top \nabla f(x_t;B_t)] \\
&= \frac{L^2\lambda^2k^2}{2}+2\mathbb{E}_{B_t,B_t'}[\nabla f(x_t;B_t)^\top \text{Proj}_G(\nabla f(x_t;B_t))] \\
&\le \frac{L^2\lambda^2k^2}{2}+2\mathbb{E}_{B_t}[\|\nabla f(x_t;B_t)\|^2] \\
&\le \frac{L^2\lambda^2k^2}{2}+2\left(\|\nabla f(x_t)\|^2+\frac{\sigma_1^2}{b}\right). \qquad (18)
\end{aligned}$$

Applying T_1 (17) and T_2 (18) to (15) yields

$$\begin{aligned}
(\eta_t-L\eta_t^2k)\|\nabla f(x_t)\|^2 \le f(x_t)-\mathbb{E}_t[f(x_{t+1})]+\eta_t M\left(\sqrt{2(\frac{\sigma_2^2}{b'}+\gamma^2)}+\frac{L\lambda k^{\frac{3}{2}}}{2}\right) \\
+\frac{L^3\eta_t^2\lambda^2k^3}{4}+\frac{L\eta_t^2k\sigma_1^2}{b}+\frac{L\eta_t^2\sigma^2C^2k}{2b^2}.
\end{aligned}$$

We choose $\eta_t=\frac{1}{2Lk}$ so that $\eta_t-L\eta_t^2k=\frac{\eta_t}{2}$. Denote $\mathbb{E}_{<t}:=\mathbb{E}_{u_{<t},z_{<t},B_{<t},B'_{<t}}$ where $u_{<t}$ is the set $\{u_0,...,u_{t-1}\}$ and similarly for $z_{<t}$, $B_{<t}$, and $B'_{<t}$. Then we have

$$\mathbb{E}_{<t}\|\nabla f(x_t)\|^2 \le 4Lk\mathbb{E}_{<t+1}[f(x_t)-f(x_{t+1})]+2M\sqrt{2\left(\frac{\sigma_2^2}{b'}+\gamma^2\right)}+L\lambda k^{\frac{3}{2}}+\frac{L^2\lambda^2k^2}{4}+\frac{\sigma_1^2}{b}+\frac{\sigma^2C^2}{2b^2}.$$

Summing up from $t=0$ to $T-1$ and dividing both sides by T yields

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{<t}[\|\nabla f(x_t)\|^2] \le \frac{4Lk}{T}\mathbb{E}_{<t+1}[f(x_0)-f(x_*)]+2M\sqrt{2\left(\frac{\sigma_2^2}{b'}+\gamma^2\right)} \\
+L\lambda k^{\frac{3}{2}}+\frac{L^2\lambda^2k^2}{4}+\frac{\sigma_1^2}{b}+\frac{\sigma^2C^2}{2b^2}.
\end{aligned}$$

□

From the upper bound, we see that when $\gamma=0$, it reduces to $O\left(\sqrt{\frac{\sigma_2^2}{b'}}+\lambda+\lambda^2+\frac{\sigma_1^2}{b}+\frac{\sigma^2}{b^2}\right)$. If we further set λ to decay, then similar as the arguments around PAZO-M bounds in Appendix B.2, the error term vanishes into $O\left(\sqrt{\frac{\sigma_2^2}{b'}}+\frac{\sigma_1^2}{b}+\frac{\sigma^2}{b^2}\right)$, which is standard for stochastic methods under a fixed learning rate, and decreases as the batch-size b, b' increase.

B.4. Convergence of PAZO-S

Theorem B.5 (Full statement of Theorem 4.3)**.** *Let the private and public data be γ-similar and Assumption 1, 2, 3, and 4 hold. For possibly non-convex f, running Algorithm 3 for T rounds using a fixed step size $\eta=\frac{1}{2L}$ gives*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{<t}[\|\nabla f(x_t)\|^2] \le \frac{4L\mathbb{E}_{<t+1}[f(x_0)-f(x_*)]}{T}+2M\left(\gamma+\frac{\sigma_2}{\sqrt{b'}}\right)+\gamma^2+\frac{\sigma_2^2}{b'}.$$

Proof. Our public data sampling process is equivalent to first sampling B_t' and then dividing it into k non-overlap partitions. We denote each partition as $B'_{t,i}, i\in[k]$. Assume that clipping does not happen, then the update rule is

$x_{t+1}-x_t=-\eta_t(g'(x_t;B'_{t,*})+\mathbb{1}(z')z')$ where $g'(x_t;B'_{t,*}):=\text{argmin}_{i\in[k+1]}f(x_t-\eta_t g(x_t;B'_{t,i});B_t)$ is the best public gradients among the k candidates and $\mathbb{1}(z')$ is an indicator variable that denotes whether adding $z'\sim\mathcal{N}(0,\epsilon^2 I_d)$ reduces the function value.

At a step t, let x_t be a fixed parameter. We apply the update to the property of L-smooth objectives and take expectation over all the randomness at this iteration, i.e., $\mathbb{E}_t:=\mathbb{E}_{u_t,z_t,B_t,B'_t}$. We have

$$\begin{aligned}
f(x_{t+1})\le& f(x_t-\eta_t g'(x_t;B'_{t,*}))\\
\le& f(x_t)-\eta_t\langle\nabla f(x_t),g'(x_t;B'_{t,*})\rangle+\frac{L\eta_t^2}{2}\|g'(x_t;B'_{t,*})\|^2\\
=& f(x_t)-\eta_t\|\nabla f(x_t)\|^2+\eta_t\langle\nabla f(x_t),\nabla f(x_t)-g'(x_t;B'_{t,*})\rangle+\frac{L\eta_t^2}{2}\|g'(x_t;B'_{t,*})\|^2\\
\le& f(x_t)-\eta_t\|\nabla f(x_t)\|^2+\eta_t\|\nabla f(x_t)\|\|\nabla f(x_t)-g'(x_t;B'_{t,*})\|+\frac{L\eta_t^2}{2}\|g'(x_t;B'_{t,*})\|^2
\end{aligned}$$

and thus

$$\begin{aligned}
\mathbb{E}_t[f(x_{t+1})]\le& f(x_t)-\eta_t\|\nabla f(x_t)\|^2+\eta_t\|\nabla f(x_t)\|\underbrace{\mathbb{E}_t[\|\nabla f(x_t)-g'(x_t;B'_{t,*})\|]}_{T_1}\\
&+\frac{L\eta_t^2}{2}\underbrace{\mathbb{E}_t[\|g'(x_t;B'_{t,*})\|^2]}_{T_2}.
\end{aligned}$$

For T_1, we note that $(\mathbb{E}_t[\|g_t-g'(x_t;B'_{t,*})\|])^2\le\mathbb{E}_t[\|g_t-g'(x_t;B'_{t,*})\|^2]\le\sigma_2^2/b'$ and thus

$$\begin{aligned}
\mathbb{E}_t[\|\nabla f(x_t)-g'(x_t;B'_{t,*})\|]\le&\mathbb{E}_t[\|\nabla f(x_t)-g_t\|]+\mathbb{E}_t[\|g_t-g'(x_t;B'_{t,*})\|]\\
\le&\gamma+\sigma_2/\sqrt{b'}.
\end{aligned}$$

For T_2, we have

$$\begin{aligned}
\mathbb{E}_t[\|g'(x_t;B'_{t,*})\|^2]=&\mathbb{E}_t[\|g'(x_t;B'_{t,*})-\nabla f(x_t)+\nabla f(x_t)\|^2]\\
\le&2\mathbb{E}_t[\|g'(x_t;B'_{t,*})-\nabla f(x_t)\|^2]+2\|\nabla f(x_t)\|^2\\
=&2\mathbb{E}_t[\|g'_t-g_t+\frac{1}{b'}\sum_{i\in B'_t}\zeta'_{t,i}\|^2]+2\|\nabla f(x_t)\|^2\\
\le&2\gamma^2+\frac{2\sigma_2^2}{b'}+2\|\nabla f(x_t)\|^2
\end{aligned}$$

Denote $\mathbb{E}_{<t}:=\mathbb{E}_{u_{<t},z_{<t},B_{<t},B'_{<t}}$ where $u_{<t}$ is the set $\{u_0,...,u_{t-1}\}$ and similarly for $z_{<t}$, $B_{<t}$, and $B'_{<t}$. Then we have

$$(\eta_t-L\eta_t^2)\mathbb{E}_{<t}\|\nabla f(x_t)\|^2\le\mathbb{E}_{<t+1}[f(x_t)-f(x_{t+1})]+\eta_t M(\gamma+\frac{\sigma_2}{\sqrt{b'}})+L\eta_t^2(\gamma^2+\frac{\sigma_2^2}{b'}).$$

Choosing $\eta_t=\frac{1}{2L}$ so that $L\eta_t^2=\eta_t/2$, summing up from $t=0$ to $T-1$, and dividing both sides by T yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{<t}[\|\nabla f(x_t)\|^2]\le\frac{4L\mathbb{E}_{<t+1}[f(x_0)-f(x_*)]}{T}+2M\left(\gamma+\frac{\sigma_2}{\sqrt{b'}}\right)+\gamma^2+\frac{\sigma_2^2}{b'}.$$

□

When $\gamma=0$, the non-vanishing error term becomes $O\left(\frac{\sigma_2}{\sqrt{b'}}+\frac{\sigma_2^2}{b'}\right)$, which is due to stochastic gradient noise.

C. Experiment Details

C.1. Datasets

The four datasets and model pairs closely follow the experiments in the existing DP literature. We provide the details of public data generation as follows.

CIFAR-10. We follow previous work (Nasr et al., 2023) that uses 4% of the training samples as public data and warm-start on the public data by training on it for a small number of epochs. Additionally, we create class imbalances among the 10 classes for public data. We treat this imbalance as a mild distribution shift from the private data. To avoid information leakage from the batchnorm layer, we start from a randomly initialized NFResNet18 (Brock et al., 2021).

Tiny-ImageNet. We follow Kurakin et al. (2022), which first pre-trains a ResNet18 on Places365 (Zhou et al., 2017) and then fine-tunes the model on Tiny-ImageNet with differential privacy. We randomly sample 4% of the Tiny-ImageNet training samples as public data, which thus comprises 20 samples per class. We use a small ViT model (10M) (Dosovitskiy et al., 2020) with random initialization.

IMDB. We follow Li et al. (2022), which uses Amazon Polarity (Zhang et al., 2015) samples as out-of-distribution (OOD) public data to guide the private learning on IMDB. We build the vocabulary based on the top 10K tokens in the IMDB training set and construct the Amazon Polarity public dataset with a size 4% of the IMDB training size, which gives us 2,000 public samples.

MNLI. We follow the few-shot setting in the past work (Malladi et al., 2023; Zhang et al., 2023) and sample 512 MNLI training examples per class. We adopt the same prompt template and start from a pre-trained RoBERTa-base model. We randomly sample 100 training examples per class from SNLI (Bowman et al., 2015) as the OOD public data.

C.2. Experiment results

We present the detailed evaluation results on the four datasets in Table $1-4$. We report the performance under multiple privacy budgets (ε,δ=$1/\#$train samples) as well as the non-private performance, which corresponds to the accuracies of SGD and MeZO. All results are obtained under the same random seed 0. Entries with '$-$' indicate failure to converge. The best accuracies are in bold and the second places are underlined.

Implementation details. For each first-order methods with public data, we vectorize the per-sample gradient computation and privatization using `vmap`. For the method with open-sourced code (GEP (Yu et al., 2021)), we adopt their provided implementation and privacy accounting.

The experiment on MNLI utilizes the codebase from Malladi et al. (2023) and Zhang et al. (2023), including their dataset processing and prompt tuning workflow. Following MeZO and DPZero, we sample the zeroth-order direction u_t from the Gaussian distribution $\mathcal{N}(0,I_d)$ in the experiments since previous work verifies that it produces very similar performance (Nasr et al., 2023; Zhang et al., 2024b; 2023) to sampling from $\sqrt{d}\mathbb{S}^{d-1}$. Similar to the first-order methods, we apply `vmap` for speedup by vectoring the q forward calls. However, given that PAZO needs smaller q's than the vanilla zeroth-order methods, we do not need to employ this memory-inefficient implementation in most settings.

PAZO-P vs. PAZO-P′. Table $1-4$ shows the performance of PAZO-P with orthonormalized public gradients (row 'PAZO-P') and with normalized public gradients (row 'PAZO-P′'). PAZO-P and PAZO-P′ have similar performance, with the deviation being $0.1\%\sim2.5\%$.

Runetime efficiency. Theoretically, we list the number of different types of operations involved in each algorithm in Table 6. Since the first-order methods require per-sample gradient computation and clipping, its number of "gradient backward", the slowest operation, is dependent on the private batch-size. This is a discouraging feature since large batch-size offers better utility/privacy tradeoffs (McMahan et al., 2017; Yu et al., 2023), creating an additional tradeoff between utility and efficiency. In contrast, the number of gradient backward steps is either 1 or $k(k\ll b)$ in zeroth-order methods. Together with the fact that the forward calls are more memory-efficient than the backward ones when vectorized, zeroth-order methods are principally more scalable.

Empirically, we evaluate the runtime in each training iteration for all the settings (Table 5). We vectorize the three settings other than the IMDB-LSTM experiment due to incompatibility between the model architecture and `vmap`. Although the

Table 1: Training NFResNet18 on CIFAR-10 from scratch.

Type	Method	ε=0.1	ε=0.5	ε=1	ε=2	ε=3	Non-private
FO	DP-SGD	46.7	49.7	50.8	54.2	54.5	86.3
FO+PUB	DPMD	64.3	66.6	67.8	68.5	69.8	
	DOPE-SGD	64.8	69.3	70.9	**73.0**	**72.9**	
	GEP	–	49.9	50.7	52.9	53.8	
ZO	DPZero	47.0	48.1	48.2	48.2	48.1	49.0
ZO+PUB (ours)	PAZO-M	**70.9**	**71.3**	**71.3**	71.2	70.5	
	PAZO-P	69.5	69.6	69.0	68.7	68.1	
	PAZO-P′	69.6	69.2	69.2	68.9	68.0	
	PAZO-S	70.3	70.3	70.2	69.8	69.7	

Table 2: Fine-tuning Places365 pre-trained ViT-small on Tiny-ImageNet.

Type	Method	ε=0.1	ε=0.5	ε=1	ε=2	Non-private
FO	DP-SGD	24.8	29.2	31.4	38.0	52.9
FO+PUB	DPMD	30.5	31.4	**34.2**	**35.5**	
	DOPE-SGD	30.7	**31.8**	32.5	34.4	
	GEP	–	30.9	30.5	31.4	
ZO	DPZero	25.1	27.6	27.5	27.9	28.6
ZO+PUB (ours)	PAZO-M	30.8	30.8	30.7	30.8	
	PAZO-P	**30.9**	31.0	31.0	31.2	
	PAZO-P′	30.7	30.8	30.8	30.9	
	PAZO-S	30.6	30.6	30.6	30.7	

Table 3: Training LSTM on IMDB from scratch.

Type	Method	ε=0.1	ε=0.5	ε=1	ε=2	ε=3	Non-private
FO	DP-SGD	50.0	66.4	69.9	73.5	75.5	89.5
FO+PUB	DPMD	71.0	72.1	73.4	76.6	76.6	
	DOPE-SGD	70.2	73.2	**75.0**	75.9	77.9	
	GEP	60.0	71.0	74.0	**77.2**	**78.6**	
ZO	DPZero	59.0	62.4	62.6	63.2	63.8	63.8
ZO+PUB (ours)	PAZO-M	73.4	73.2	74.5	73.2	73.6	
	PAZO-P	71.0	73.7	73.2	73.0	72.7	
	PAZO-P′	69.4	69.8	70.7	70.0	70.5	
	PAZO-S	**74.6**	**74.2**	73.8	73.9	74.2	

Table 4: Prompt-tuning RoBERTa-base on MNLI.

Type	Method	ε=0.1	ε=0.5	ε=1	ε=2	ε=3	Non-private
FO	DP-SGD	52.6	59.3	63.5	68.4	72.0	78.9
FO+PUB	DPMD	56.5	67.0	68.1	**71.5**	**72.8**	
	DOPE-SGD	59.7	67.2	68.0	70.1	72.5	
	GEP	–	–	–	–	–	
ZO	DPZero	–	55.2	58.2	60.4	62.6	68.4
ZO+PUB (ours)	PAZO-M	67.1	67.3	67.8	67.7	67.5	
	PAZO-P	63.5	68.3	**69.8**	69.7	70.3	
	PAZO-P′	61.0	68.1	68.8	69.0	69.4	
	PAZO-S	**68.2**	**68.6**	68.9	68.6	69.0	

Table 5: Speed of each method on different datasets (in s/iter). It shows that PAZO offers up to 16× runtime speedup per training iteration compared to the baselines. All numbers are averaged over 20 iterations.

	CIFAR-10	Tiny-ImageNet	IMDB	MNLI
DP-SGD	0.420	0.366	0.173	1.697
DPMD	0.462	0.404	0.183	1.761
DOPE-SGD	0.424	0.365	0.172	2.187
GEP	0.830	0.548	0.252	–
DPZero	0.081	0.132	0.016	1.934
PAZO-M	0.051	0.073	0.019	0.852
PAZO-P	0.149	0.168	0.042	1.244
PAZO-S	0.102	0.142	0.019	1.118
Speedup	16×	7×	15×	2×

MNLI experiments enjoys only 2× of speedup by using PAZO, Malladi et al. (2023) shows that zeroth-order methods will be significantly faster as the model scales up.

C.3. Hyperparameter tuning

This section presents our hyperparameter search grid and the results of our methods under different hyperparameter values.

Hyperparameter selection. For all the first-order methods and PAZO, we set the number of epochs to 100. Since the vanilla zeroth-order methods benefit from training for more iterations (Zhang et al., 2023; Malladi et al., 2023), we try training for 100, 200, and 300 epochs with their corresponding correct noise multiplier σ applied. Due to increased noise added when more epochs are allowed, we observe that the epoch number of 200 produces the best performance across settings. We thus train for 200 epochs in all DPZero experiments. The values of the smoothing parameter λ are presented in Table 7.

Sensitivity to q. Table 8 shows that the performance of the vanilla private zeroth-order method relies on setting $q{>}1$, which slows down the training and harms utility due to increased noise added for privatization. In contrast, PAZO is less dependent on increased q due to the assistance from public data. This implies that PAZO has approximately the same workload of hyperparameter tuning as DPZero: Under a reasonable or intuitive choice of the hyperparameters for public data sampling, one only needs to find a good combination of clipping norm C and learning rate η.

Sensitivity to introduced hyperparameters. Apart from Figure 6, we also present the hyperparameter sensitivity study on the other two datasets Tiny-ImageNet and IMDB in Figure 8. The conclusion is the same as in the main text: PAZO is not sensitive to the values of the introduced hyperparameters.

Table 6: The number of different operations per iteration of each method.

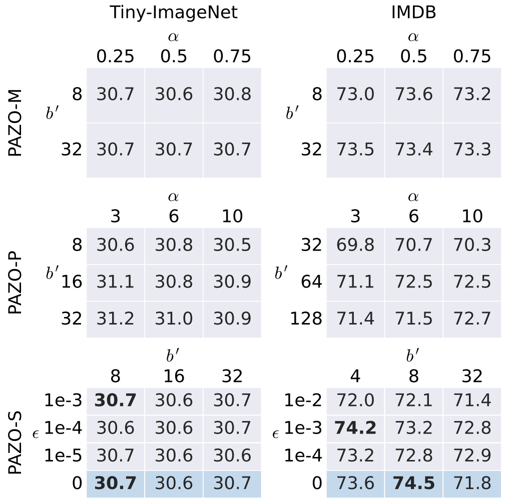
	# Private forward	# Public for+backward	# Private backward
DP-SGD	b	–	b
DPMD	b	1	b
DOPE-SGD	b	1	b
GEP	b	b'	b
DPZero	$2q$	–	–
PAZO-M	$2q$	1	–
PAZO-P	$2q$	k	–
PAZO-S	$k{+}1$	k	–

Table 7: Values of the smoothing parameter λ in each experiment.

	CIFAR-10	Tiny-ImageNet	IMDB	MNLI
MeZO	10^{-2}	10^{-2}	10^{-2}	10^{-3}
DPZero	10^{-2}	10^{-2}	10^{-2}	10^{-3}
PAZO-M	10^{-2}	10^{-2}	10^{-2}	10^{-3}
PAZO-P	10^{-2}	10^{-2}	10^{-1}	10^{-2}

Table 8: Performance vs. q in different settings. In each cell, the first row represents the accuracy under $q{=}1$ and the second represents that under $q{=}5$. We observe that DPZero benefits from increased q by 1.0%, 2.4%, 4.8%, and 7.2% accuracy points on four datasets. In contrast, PAZO has stable performance under different q.

$\frac{q=1}{q=5}$	CIFAR-10	Tiny-ImageNet	IMDB	MNLI
DPZero	47.1 48.1	25.5 27.9	59.0 63.8	55.4 62.6
PAZO-M	70.1 70.3	30.8 30.8	72.9 73.6	67.5 68.3
PAZO-P	68.1 68.6	31.2 31.0	72.7 72.7	68.6 70.9



Tiny-ImageNet

PAZO-M b' \ α	0.25	0.5	0.75
8	30.7	30.6	30.8
32	30.7	30.7	30.7

PAZO-P b' \ α	3	6	10
8	30.6	30.8	30.5
16	31.1	30.8	30.9
32	31.2	31.0	30.9

PAZO-S ϵ \ b'	8	16	32
1e-3	**30.7**	30.6	30.7
1e-4	30.6	30.6	30.7
1e-5	30.7	30.6	30.6
0	**30.7**	30.6	30.7

IMDB

PAZO-M b' \ α	0.25	0.5	0.75
8	73.0	73.6	73.2
32	73.5	73.4	73.3

PAZO-P b' \ α	3	6	10
32	69.8	70.7	70.3
64	71.1	72.5	72.5
128	71.4	71.5	72.7

PAZO-S ϵ \ b'	4	8	32
1e-2	72.0	72.1	71.4
1e-3	**74.2**	73.2	72.8
1e-4	73.2	72.8	72.9
0	73.6	**74.5**	71.8

Figure 8: All PAZO methods are robust to different values of their introduced hyperparameters. Each number represents the best accuracy after the standard hyperparameters for zeroth-order private optimization (C and η) are tuned. Blue cells are for PAZO-S's performance without having a noisy candidate.

Influence of ϵ in PAZO-S. Figure 6 and Figure 8 show that the performance of PAZO-S is robust to different ϵ values. Since having no noisy candidate is equivalent to setting $\epsilon=0$, we compare the best performance of having a noisy candidate (purple cells) with none (blue cells). The conclusion is consistent: Having $\epsilon \neq 0$ offers the opportunity to improve performance in general, but it does not harm significantly to leave it less tuned.

Table 9: The hyperparameter search grid for CIFAR-10 and Tiny-ImageNet.

Algorithm		CIFAR-10	Tiny-ImageNet
SGD	η	{0.01, 0.02, 0.05, 0.1, 0.2, 0.5}	{0.001, 0.005, 0.01, 0.05, 0.1}
	b	{8, 32, 64}	{64}
DP-SGD	b	{0.01, 0.02, 0.05, 0.1, 0.2}	{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0}
	C	{0.1, 0.5, 1.0, 2.0}	{0.01, 0.1, 0.5, 1.0, 2.0}
DOPE-SGD	η	{0.01, 0.02, 0.05, 0.1, 0.2}	{0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2}
	b'	{8, 32, 128}	{8, 32, 128}
	C	{0.1, 0.5, 1.0, 2.0}	{0.1, 0.5, 1.0, 2.0, 4.0}
DPMD	η	{0.02, 0.05, 0.1, 0.2, 0.5}	{0.005, 0.01, 0.02, 0.05, 0.1, 0.2}
	b'	{8, 32, 128}	{8, 32, 128}
	C	{0.1, 0.5, 1.0, 2.0}	{0.01, 0.1, 0.5, 1.0, 2.0}
GEP	η	{0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5}	{0.01, 0.02, 0.05, 0.1, 0.2, 0.5}
	b'	{8, 32, 128}	{8, 32, 128}
	C_1	{0.1, 0.5, 1.0, 2.0}	{0.1, 0.5, 1.0, 1.5, 2.0}
MeZO	η	{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1}	{1e-4, 2e-4, 5e-4, 1e-3, 2e-3}
	b	{64}	{64}
DPZero	η	{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0}	{1e-4, 2e-4, 5e-4, 1e-3, 2e-3}
	C	{1.0}	{1.0}
PAZO-M	η	{0.1, 0.2, 0.5}	{1e-5, 2e-5, 5e-5, 1e-4, 2e-4, 5e-4}
	b'	{8, 32}	{8, 32}
	α	{0.25, 0.5, 0.75}	{0.25, 0.5, 0.75}
	C	{1.0}	{1.0}
PAZO-P	η	{0.2, 0.5, 1.0, 1.5, 2.0}	{0.2, 0.5, 1.0, 1.5, 2.0}
	b'	{8, 16, 32}	{8, 16, 32}
	k	{3, 6, 10}	{3, 6, 10}
	C	{0.5, 1.0, 2.0}	{0.5, 1.0, 2.0}
PAZO-S	η	{0.01, 0.02, 0.05, 0.1, 0.2}	{0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2}
	b'	{8, 16, 32}	{8, 32, 128}
	k	{3}	{3}
	ϵ	{0.01, 0.001}	{0.001, 0.0001}
	C	{0.5, 1.0, 2.0, 4.0}	{0.5, 1.0, 2.0, 4.0}

Table 10: The hyperparameter search grid for IMDB and MNLI.

Algorithm		IMDB	MNLI
SGD	η	{0.1, 0.2, 0.5, 1.0, 1.5}	{1e-6, 1e-5, 1e-4, 1e-3, 5e-3, 1e-2}
	b	{64}	{8, 32, 64}
DP-SGD	b	{0.01, 0.02, 0.05, 0.1, 0.2, 0.1}	{2e-6, 5e-6, 1e-5, 2e-5, 5e-5, 1e-4}
	C	{0.1, 0.5, 1.0, 2.0, 4.0}	{10, 20, 50, 100, 150, 200, 250}
DOPE-SGD	η	{0.005, 0.01, 0.02, 0.05, 0.1}	{5e-6, 1e-5, 2e-5, 5e-5, 1e-4}
	b'	{8, 32, 128}	{8. 32}
	C	{0.1, 0.5, 1.0, 2.0, 4.0}	{10, 20, 50, 100, 150, 200, 250}
DPMD	η	{0.005, 0.01, 0.02, 0.05, 0.1}	{2e-6, 5e-6, 1e-5, 2e-5, 5e-5, 1e-4, 2e-4}
	b'	{8, 32, 128}	{8, 32}
	C	{0.1, 0.5, 1.0, 2.0, 4.0}	{10, 20, 50, 100, 150, 200, 250}
GEP	η	{0.01, 0.02, 0.05, 0.1}	{2e-6, 5e-6, 1e-5, 2e-5, 5e-5, 1e-4, 2e-4}
	b'	{8, 32}	{8, 32}
	C_1	{0.1, 0.5, 1.0, 2.0}	{10, 20, 50, 100, 150, 200, 250}
MeZO	η	{0.002, 0.005, 0.01, 0.02, 0.05, 0.1}	{1e-7, 1e-6, 2e-6, 5e-6, 1e-5, 1e-4}
	b	{64}	{64}
DPZero	η	{0.002, 0.005, 0.01, 0.02, 0.05, 0.1}	{1e-6, 2e-6, 5e-6, 1e-5, 2e-5, 5e-5}
	C	{0.1, 0.5, 1.0, 2.0}	{10, 20, 50, 100, 150, 200, 250}
PAZO-M	η	{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0}	{1e-4, 2e-4, 5e-4, 1e-3, 2e-3}
	b'	{8, 32}	{8, 32}
	α	{0.25, 0.5, 0.75}	{0.25, 0.5, 0.75}
	C	{0.1, 0.5, 1.0, 2.0, 4.0}	{10, 20, 50, 100, 150, 200, 250}
PAZO-P	η	{0.1, 0.2, 0.5, 1.0, 1.4, 2.0}	{5e-5, 1e-4, 2e-4, 5e-4, 1e-3, 2e-3}
	b'	{32, 64, 128}	{8, 16, 32}
	k	{3, 6, 10}	{3, 6, 10}
	C	{0.5, 1.0, 2.0, 4.0}	{10, 20, 50, 100, 150, 200, 250}
PAZO-S	η	{0.1, 0.2, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0}	{1e-4, 2e-4, 5e-4, 1e-3, 2e-3, 5e-3}
	b'	{8, 32, 128}	{8, 32}
	k	{3}	{3}
	ϵ	{0.01, 0.001}	{0.01, 0.001}
	C	{0.1, 0.5, 1.0}	{0.1, 0.5, 1.0}