Private Zeroth-Order Optimization with Public Data

Xuchen Gong **Tian Li**
University of Chicago
{xuchengo,litian}@uchicago.edu

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 have only been evaluated in 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 superior privacy/utility tradeoffs across vision and text tasks in both pre-training and fine-tuning settings, outperforming the best first-order baselines (with public data) especially in highly private regimes, while offering up to $16\times$ runtime speedup.

1 Introduction

Differentially private (DP) is a widely-used framework to protect sensitive information so that adversaries cannot infer if any user or sample participates in the computation. When applied to machine learning tasks, popular DP algorithms based on privatizing first-order gradients (such as DP-SGD [1]) 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 extra memory for computation [2, 3].

To tackle this, zeroth-order optimization offers an attractive alternative for DP training, as it leverages function queries (scalar values) to approximate the gradients and is hence inherently amenable to privatization [4, 5]. However, randomly searching in a potentially high-dimensional space based on function query feedback can be rather inefficient [4]. Prior work has demonstrated competitive performance of (private) zeroth-order methods only in the limited context of language model fine-tuning with prompts [6, 7, 8, 9, 10] or models with extreme sparsity [11]. In addition, there is still a utility gap between private zeroth-order and first-order approaches on challenging tasks [8].

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 non-sensitive public data, whose batch gradients provide informative guidance on perturbing the parameter space. We thus introduce PAZO, a suite of zeroth-order DP algorithms that leverage a small amount of public data with similar distributions as private data along with 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,

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

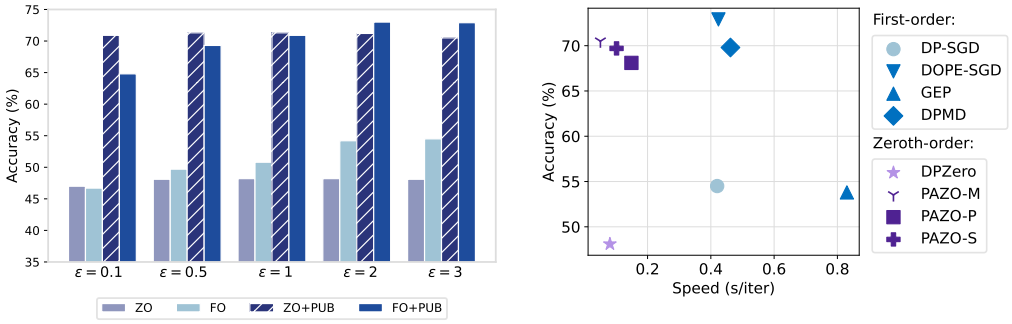

Figure 1: Results of CIFAR-10 with NFResNet18 trained from scratch under privacy budget $\varepsilon = 3$. *Left:* Zeroth-order methods demonstrate consistent accuracies under various privacy budgets compared with the best first-order method with public data. *Right:* Proposed zeroth-order approaches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

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 tuning with prompts, we investigate both image and text 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 [1]), whereas even modest amounts of public data can significantly close the gap, especially in highly private regimes. In particular, the best zeroth-order method with public data can match or even outperform the best public-data-assisted first-order counterpart, 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 summarized 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, all with improved convergence rate in terms of model dimension d. PAZO-M improves the vanilla zeroth-order method by a factor of $\log d$, and PAZO-{P,S} obtain d-independent rates.
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 Related Work and Preliminaries

Differential privacy. In this work, we focus on the popular definition of sample-level DP [1, 12].

Definition 1 (Differential privacy [12])**.** *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 [13]. Extensive prior research has been proposed

to improve privacy/utility tradeoffs, including increasing the batch size [14, 15], using public or side information [16, 17, 18], and reducing the dimensionality of gradients [19]. Another bottleneck of deploying DP algorithms at scale lies in the computation (or memory) cost [2]. For example, vanilla DP-SGD computes and stores per-sample clipped gradients, leading to memory consumption $O(bd)$ where b is the private batch size and d is the model dimension. Existing methods, such as ghost-clipping/bookkeeping [20], reduce layer-wise gradient storage to $\min\{2bp, bd\}$, where d is the layer dimension and p is the feature dimension of this layer, i.e., sequence length for text data. 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 [21, 22, 23], hyperparameter tuning [24], and data-driven science workloads [25]. One fundamental challenge of zeroth-order methods is the need for a large number of function queries to reduce the variance of the estimate [4]. Existing work has explored various techniques to improve the estimate, such as incorporating the previous estimated gradient directions [26] and sparsifying gradients [11]. Our work focuses on private training, and the proposed techniques can be combined with those 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 [4], involves two evaluations of function values:

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

where ξ 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 (1) $f_\lambda(x)$ is differentiable and (2) $\mathbb{E}_u[g_\lambda(x; \xi_i)] = \nabla f_\lambda(x)$ [4, 8]. 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 samples as opposed to mini-batches. Therefore, recent work has considered privatizing zeroth-order algorithms [8, 7, 27, 28] 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)), the private zeroth-order gradient $\tilde{g}_\lambda(x; B)$ is computed by

$$\tilde{g}_\lambda(x; B) := \left(\frac{1}{b} \sum_{\xi \in B} \text{clip}_C \left(\frac{f(x + \lambda u; \xi) - f(x - \lambda u; \xi)}{2\lambda} \right) + z \right) u, \tag{2}$$

where $b = |B|$ is batch size, $z \sim \frac{1}{b}\mathcal{N}(0, C^2\sigma^2)$ is privacy noise, and u is a random direction, e.g., sampled uniformly from a sphere $\sqrt{d}\mathbb{S}^{d-1}$. We can query the raw data multiple times per iteration by sampling multiple u's to improve the estimate (Section 3). Prior private zeroth-order work mostly focuses on language model tuning with prompts, and additionally there still exists a big performance gap between zeroth- and first-order methods [8, 7, 27]. 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 Private Zeroth-Order Optimization

Given zeroth-order oracles on private data and first-order oracles on public data, we aim to blend public gradients 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 using this public prior that significantly outperform zeroth-order baselines without public data and result in competitive/superior performance relative to DP-SGD with public data. We analyze their 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)). 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. The updating rule is summarized in Algorithm 1 below.

We note that the norm of two-point gradient estimates is approximately d times that of the true private gradient [6], so it is important to align their norms so that tuning the mixing coefficient can be easier. To achieve this, we sample u uniformly from the sphere $r\mathbb{S}^{d-1}$ with radius $r = d^{\frac{1}{4}}$ so that $\mathbb{E}_{u_t}[\|g_\lambda(x)\|^2] \approx \|\nabla f(x)\|^2$. The proof is detailed in Appendix A. The mixing coefficient α can be adjusted to change the emphasis on the public gradient. Although α is an introduced hyperparameter, as shown in experiments (Section 5), PAZO-M is robust to a wide range of α values in $(0, 1)$ as well as the public batch size b', as long as the L_2 norms of $g_{\texttt{pub}}$ and $\tilde{g}/q$ are aligned.

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 [29], PAZO-M differs from this work in terms of the effective optimization objectives, bias-variance tradeoffs, analyses, and application settings.

Algorithm 1 PAZO-M

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

3.2 PAZO-P: Sampling in Public Gradient Subspace

Recall that the two-point estimator samples perturbations u in the sphere $\sqrt{d}\mathbb{S}^{d-1}$. Such random exploration along two directions λu and $-\lambda u$ can result in a loose estimation of the real gradients in high-dimensional settings. In this section, we assume the true gradient on private data is close to the space formed by public gradients. Based on this assumption, we constrain the private gradient estimates to lie 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). This gives us a much lower-dimensional optimization problem.

Formally, suppose we have access to k ($k \ll d$) mini-batch stochastic gradients obtained on public data. 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 $\sqrt{k}\mathbb{S}^{k-1}$. We propose the following update rule (sampling only one u as an example) in the non-private case:

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

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, $g_\lambda^G(x;\xi)$ estimates the orthogonal projection of the true gradient onto the public gradient subspace when $\lambda \to 0$, i.e.,

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

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

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

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 [30, 31].

Previous work proposes constraining the random search to the principal components of surrogate gradients [32]. PAZO-P differs from theirs in allowing to use non-orthonormalized G. Section 5 presents the performance of PAZO-P with orthonormalization, and the complete results in Tables 1-4 demonstrate the 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 $\{g_1, \ldots, g_k\}$, and model parameter $x \in \mathbb{R}^d$, we have that

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

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. Inspired by this observation, 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 5-8 in Algorithm 3). Considering the residual error between

Algorithm 3 PAZO-S

Input: Same as Algorithm 2, and perturbation scale ϵ
for $t = 0, \cdots, T-1$ **do**
  Sample a mini-batch $B(|B| = b)$ of private training data $\{\xi_1, ..., \xi_b\}$
  Sample k mini-batches of public data and obtain their gradients $\{g_1, ..., g_k\}$
  for $j = 1, ..., k$ **do**
    $f_j \leftarrow \frac{1}{b} \sum_{i=1}^{b} \text{clip}_C \left(f(x_t - \eta g_j; \xi_i) \right) + z$ where $z \sim \frac{1}{b}\mathcal{N}(0, (k+1)C^2\sigma^2)$
  end for
  $\hat{j} \leftarrow \arg\min_{j \in [k]} f_j$
  $g_{k+1} \leftarrow g_{\hat{j}} + z'$ where $z' \sim \mathcal{N}(0, \epsilon^2 I_d)$
  $f_{k+1} \leftarrow \frac{1}{b} \sum_{i=1}^{b} \text{clip}_C \left(f(x_t - \eta g_{k+1}; \xi_i) \right) + z$ where $z \sim \frac{1}{b}\mathcal{N}(0, (k+1)C^2\sigma^2)$
  $j^* \leftarrow \arg\min_{j \in [k+1]} f_j$
  $x_{t+1} \leftarrow x_t - \eta g_{j^*}$
end for

the public and private subspace, we create an additional noise vector z' (Line 9), 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 11). 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 [33]. Applying standard moments accountant method [1] 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 2 (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.*

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 [16]. Together with their assumption on the bounded gradient norm, their similarity condition implies ours and is thus a stronger assumption. Next, we present additional assumptions.

Assumption 1. *$f(x;\xi)$ is M-Lipschitz for any $x \in \mathbb{R}^d$ and any subset data ξ*

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*

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

Additionally, let c_1 and c_2 be the constants that make PAZO-M satisfy (ε,δ)-differential privacy for any $\varepsilon < c_1 b^2 T/n^2, \delta > 0$. Then PAZO-M obtains the error rate

$$O\left(\frac{1-\alpha}{\alpha}\sqrt{d}\right) + O\left(\gamma^2 \frac{\alpha\sqrt{d}}{2(1-\alpha)+\alpha\sqrt{d}} + \frac{\sigma_1^2}{b}\frac{(1-\alpha)\sqrt{d}}{(1-\alpha)\sqrt{d}+\alpha} + \frac{\sigma_2^2}{b'}\frac{\alpha^2\sqrt{d}}{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)}\right)$$

by choosing the parameters

$$\eta = \frac{2(1-\alpha)+\alpha\sqrt{d}}{4L((1-\alpha)^2\sqrt{d}+\alpha(1-\alpha))}, \quad \lambda \leq \frac{1}{Ld^{\frac{5}{4}}}, \quad C = 1+\sqrt{2}d^{\frac{1}{4}}M, \quad \textit{and}$$

$$T = \frac{4n\varepsilon[(1-\alpha)\sqrt{d}+\alpha]}{c_2 C[2(1-\alpha)+\alpha\sqrt{d}]}\sqrt{\frac{2L[f(x_0)-f(x_*)]}{\sqrt{d}\log(1/\delta)}}.$$

We present several discussions on the results. First, we see that the first term in the error rate has dependence $O(\frac{1-\alpha}{\alpha}\sqrt{d})$, which saves a factor of $\log d$ compared to DPZero, together with a constant improvement if $\alpha > \frac{1}{2}$. Due to the usage of biased public gradients, we additionally have an error $O(\gamma^2\alpha\sqrt{d} + \sigma_2^2\alpha^2\sqrt{d}/b')$, which decreases to 0 as α decreases to 0. Second, there is a term related to the variance of the stochastic gradients σ_1^2/b, which is standard when we assume constant learning rates [34] and would reduce as the batch size b increases. Third, we provide a conservative upper bound by choosing the clipping threshold C larger than needed. We can also naturally extend our current analysis to incorporate more advanced clipping analysis [8].

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*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(x_t)\|^2] \leq O\left(\frac{1}{T}\right) + O\left(\sqrt{\gamma^2 + \frac{\sigma_2^2}{b'}} + \frac{\sigma_1^2}{b} + \frac{\sigma^2}{b^2}\right). \tag{5}$$

Additionally, let c_1 and c_2 be the constants that make PAZO-P satisfy (ε, δ)-differential privacy for any $\varepsilon < c_1 b^2 T/n^2, \delta > 0$. Then PAZO-P obtains the error rate

$$O(k) + O\left(\sqrt{\gamma^2 + \frac{\sigma_2^2}{b'}} + \frac{\sigma_1^2}{b}\right)$$

by choosing the parameters

$$\eta = \frac{1}{2Lk}, \quad \lambda \leq \frac{1}{Lk^{\frac{3}{2}}}, \quad C = 1 + \sqrt{2k}M, \quad \text{and } T = \frac{n\varepsilon}{c_2 C}\sqrt{\frac{8Lk[f(x_0) - f(x_*)]}{\log(1/\delta)}}.$$

This shows that we have d-independent error rate $O(k)$, with the dimension of the subspace k being small a constant $k \ll \log d$ in practice. We additionally have the error term $O(\gamma^2 + \sigma_2^2/b')$ from the biased stochastic public gradients and $O(\sigma_1^2/b)$ from the stochastic private gradients.

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*

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

This allows us to take $T \to \infty$, $\eta = 1/(4L)$, and $\epsilon \leq 1/\sqrt{d}$ to achieve a d-independent error bound $O(\gamma^2 + \sigma_2^2/b')$. When γ approaches zero, the remaining term σ_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} across both vision and language domains, and 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 follow the experiments in the existing DP literature, including (1) Training NFResNet18 on CIFAR-10 [35] from scratch, (2) fine-tuning Places365 pre-trained ViT-S on Tiny-ImageNet [36], (3) training LSTM on IMDB [37] from scratch, and (4) fine-tuning RoBERTa-base with prompts on MNLI [38]. We introduce distribution shifts between private and public data, such as class imbalance and semantic context shifts of various extents. The details of public data generation and the impact of different public data distribution shifts on algorithm performance and γ-similarity values are presented in Appendix C.1.

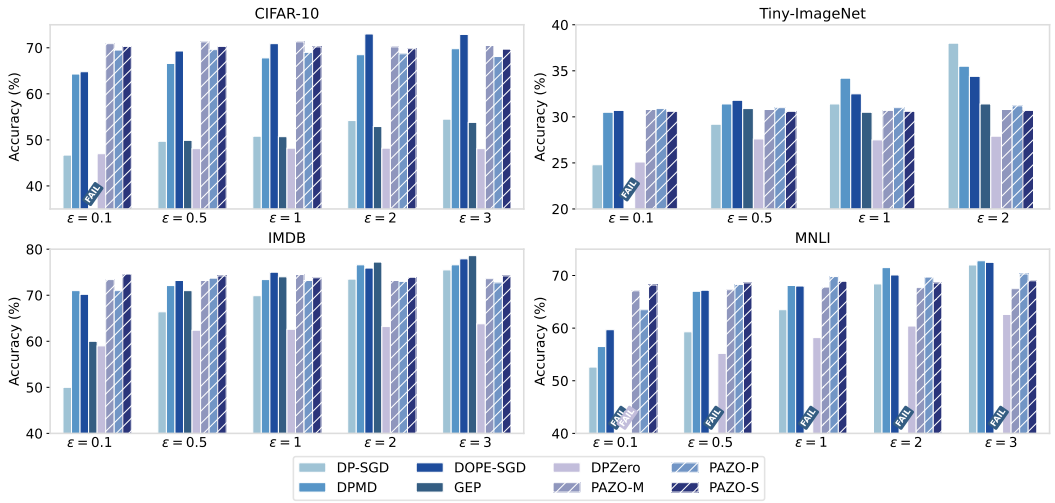

Figure 2: Performance of PAZO and the baselines in four settings. It shows that (1) all three PAZO variants outperform DPZero across all datasets, (2) all of the first-order methods (DP-SGD, DPMD, DOPE-SGD, and GEP), with or without public data, are more sensitive to smaller ε's than zeroth-order ones, 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.

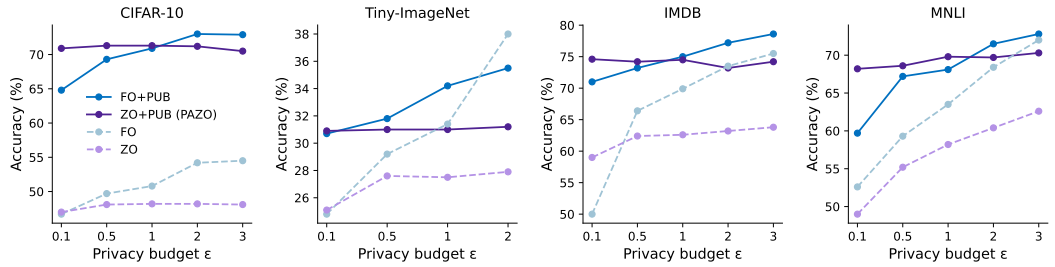

Figure 3: We compare the best private zeroth-order (ZO) methods with the best private first-order (FO) methods, with public data (+PUB) or without. Note that ZO+PUB is PAZO. It shows that (1) with or without public data, the performance gap between ZO and FO decreases as ε decreases, (2) using public 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.

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 [1], the plain first-order method without public data, (2) DPZero [8], the plain zeroth-order method without public data, and (3) the state-of-the-art first-order algorithms with public data, including DPMD [39], GEP [40], and DOPE-SGD [41].

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 performance of the best PAZO variant among three (denoted as 'ZO+PUB') and performance of the best public-data-augmented first-order method (denoted as 'FO+PUB') under

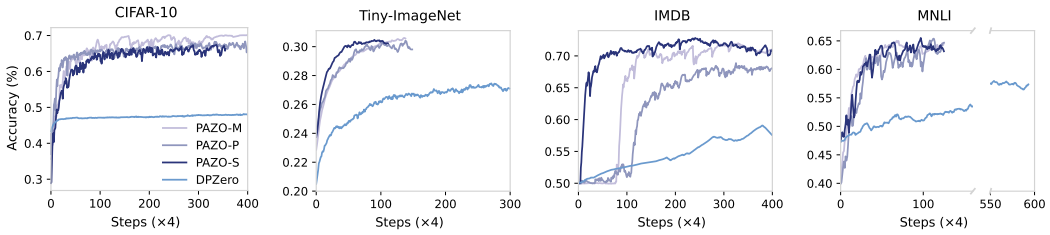

Figure 4: Convergence speed of private zeroth-order methods with (PAZO) or without (DPZero) public data. We observe that PAZO variants have slightly different convergence speed, but they are all consistently faster than the baseline. The reported are smoothed test accuracies under privacy $\varepsilon = 1$.

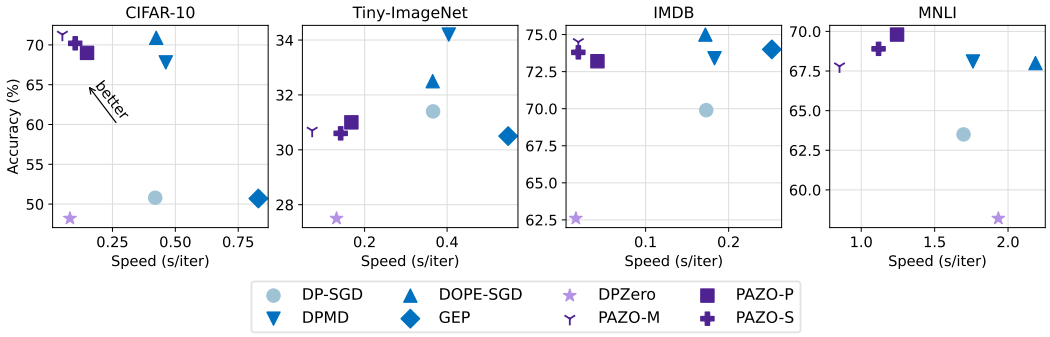

Figure 5: The utility/speed tradeoffs of different methods. It shows that PAZO is up to 16× faster in each training iteration than FO and FO+PUB while being comparably performant. The reported results are under privacy budget $\varepsilon = 1$, and the detailed numbers are in Table 7.

different ε's in Figure 3. It shows that although vanilla zeroth-order (ZO) may underperform first-order (FO) methods, if we augment both with public data, PAZO performs comparably or even superior to the best first-order approach with public data (FO+PUB), while being more memory-efficient.

5.3 Time Efficiency

In this section, we present the time efficiency of PAZO. It is faster than private first-order methods (with or without public data) as it does not require per-sample gradient clipping, and it also converges faster than private zeroth-order baselines.

#Iterations to converge. MeZO and DPZero present results with zeroth-order methods running $100\times$ and $10\times$ more steps than first-order ones [6, 8], but PAZO converges much faster due to assistance from public data. Figure 4 plots the convergence speed of DPZero and PAZO-{M, P, S}, illustrating that public information significantly accelerates the convergence of (private) zeroth-order methods. This property is particularly favorable to differentially private training as smaller accumulative noise would be added due to fewer iterations needed to converge.

Runtime per iteration. Theoretically, we compare the number of different operations in each method in Table 8. Since the number of forward and backward passes in first-order methods depends on the private batch size, first-order methods can be dramatically slow since large-batch training is favorable in DP [14, 42]. 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 a fair comparison, we adopt optimized implementations to speed up first-order DP algorithms, including vectorization, just-in-time compilation, and static graph optimization [2]. 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 $2 \sim 16\times$ faster in each training iteration.

5.4 Robustness to Hyperparameters

We have each method's hyperparameters tuned via grid search, and the detailed grid values are in Appendix C.3. Zeroth-order methods sample q random 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 shown in Table 10, DPZero benefits from increased q for improved accuracy, while PAZO has reduced dependence on q's due to the guidance from public information.

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

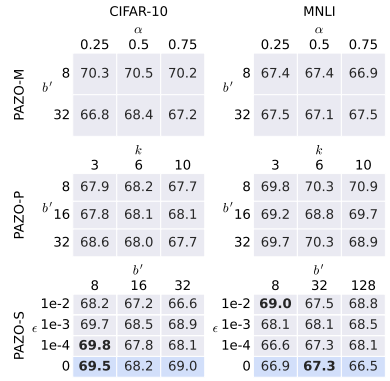

Figure 6: PAZO is non-sensitive to their introduced hyperparameters. Each number represents the best accuracy after the standard hyperparameters for zeroth-order private optimization (C and η) are tuned. Blue cells indicate PAZO-S performance w/o a noisy candidate.

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.

Acknowledgement

We thank Kamalika Chaudhuri, Chuan Guo, Saeed Mahloujifar, and Manzil Zaheer for helpful discussions at early stages of this project. We acknowledge the NAIRR Pilot program and AWS for contributing cloud credits to support this research project.

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 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Pranav Subramani, Nicholas Vadivelu, and Gautam Kamath. Enabling fast differentially private sgd via just-in-time compilation and vectorization. *Advances in Neural Information Processing Systems*, 34:26409–26421, 2021.

[3] Sebastian Rodriguez Beltran, Marlon Tobaben, Joonas Jälkö, Niki Andreas Loppi, and Antti Honkela. Towards efficient and scalable implementation of differentially private deep learning.

[4] 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.

[5] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952.

[6] 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.

[7] 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.

[8] Liang Zhang, Bingcong Li, Kiran Koshy Thekumparampil, Sewoong Oh, and Niao He. Dpzero: Private fine-tuning of language models without backpropagation. *arXiv preprint arXiv:2310.09639*, 2023.

[9] Shaocong Ma and Heng Huang. Revisiting zeroth-order optimization: Minimum-variance two-point estimators and directionally aligned perturbations. In *The Thirteenth International Conference on Learning Representations*.

[10] Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D Lee, Wotao Yin, Mingyi Hong, et al. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark. *arXiv preprint arXiv:2402.11592*, 2024.

[11] Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Jiancheng Liu, Konstantinos Parasyris, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. Deepzero: Scaling up zeroth-order optimization for deep model training. *arXiv preprint arXiv:2310.02025*, 2023.

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

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

[14] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[15] Tom Sander, Yaodong Yu, Maziar Sanjabi, Alain Durmus, Yi Ma, Kamalika Chaudhuri, and Chuan Guo. Differentially private representation learning via image captioning. *arXiv preprint arXiv:2403.02506*, 2024.

[16] Tian Li, Manzil Zaheer, Sashank Reddi, and Virginia Smith. Private adaptive optimization with side information. In *International Conference on Machine Learning*, pages 13086–13105. PMLR, 2022.

[17] Hilal Asi, John Duchi, Alireza Fallah, Omid Javidbakht, and Kunal Talwar. Private adaptive gradient methods for convex optimization. In *International Conference on Machine Learning*, pages 383–392. PMLR, 2021.

[18] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

[19] Yingxue Zhou, Zhiwei Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private sgd with gradient subspace identification. *arXiv preprint arXiv:2007.03813*, 2020.

[20] 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.

[21] 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 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[22] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pages 2137–2146. PMLR, 2018.

[23] Astha Verma, AV Subramanyam, Siddhesh Bangar, Naman Lal, Rajiv Ratn Shah, and Shin'ichi Satoh. Certified zeroth-order black-box defense with robust unet denoiser. *arXiv preprint arXiv:2304.06430*, 2023.

[24] Bin Gu, Guodong Liu, Yanfu Zhang, Xiang Geng, and Heng Huang. Optimizing large-scale hyperparameters via automated learning algorithm. *arXiv preprint arXiv:2102.09026*, 2021.

[25] Samuel C Hoffman, Vijil Chenthamarakshan, Kahini Wadhawan, Pin-Yu Chen, and Payel Das. Optimizing molecules using efficient queries from property evaluations. *Nature Machine Intelligence*, 4(1):21–31, 2022.

[26] Florian Meier, Asier Mujika, Marcelo Matheus Gauy, and Angelika Steger. Improving gradient estimation in evolutionary strategies with past descent directions. *arXiv preprint arXiv:1910.05268*, 2019.

[27] 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.

[28] Qinzi Zhang, Hoang Tran, and Ashok Cutkosky. Private zeroth-order nonsmooth nonconvex optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=IzqZbNMZOM`.

[29] Zeman Li, Xinwei Zhang, Peilin Zhong, Yuan Deng, Meisam Razaviyayn, and Vahab Mirrokni. Addax: Utilizing zeroth-order gradients to improve memory efficiency and performance of sgd for fine-tuning language models. *arXiv preprint arXiv:2410.06441*, 2024.

[30] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[31] Francesco Croce, Sylvestre-Alvise Rebuffi, Evan Shelhamer, and Sven Gowal. Seasoning model soups for robustness to adversarial and natural distribution shifts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12313–12323, 2023.

[32] Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In *International Conference on Machine Learning*, pages 4264–4273. PMLR, 2019.

[33] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(1):3–37, 2022.

[34] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

[35] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

[36] mnmoustafa and Mohammed Ali. Tiny imagenet. `https://kaggle.com/competitions/tiny-imagenet`, 2017. Kaggle.

[37] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-1015`.

[38] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[39] Ehsan Amid, Arun Ganesh, Rajiv Mathews, Swaroop Ramaswamy, Shuang Song, Thomas Steinke, Vinith M Suriyakumar, Om Thakkar, and Abhradeep Thakurta. Public data-assisted mirror descent for private model training. In *International Conference on Machine Learning*, pages 517–535. PMLR, 2022.

[40] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. *arXiv preprint arXiv:2102.12677*, 2021.

[41] Milad Nasr, Saeed Mahloujifar, Xinyu Tang, Prateek Mittal, and Amir Houmansadr. Effectively using public data in privacy preserving machine learning. In *International Conference on Machine Learning*, pages 25718–25732. PMLR, 2023.

[42] Yaodong Yu, Maziar Sanjabi, Yi Ma, Kamalika Chaudhuri, and Chuan Guo. Vip: A differentially private foundation model for computer vision. *arXiv preprint arXiv:2306.08842*, 2023.

[43] Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International conference on machine learning*, pages 1059–1071. PMLR, 2021.

[44] Alexey Kurakin, Shuang Song, Steve Chien, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at imagenet scale with differential privacy, 2022. *URL https://arxiv.org/abs/2201.12328*, 2022.

[45] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[46] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[47] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[48] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[49] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

A Algorithm Details

A.1 PAZO-M Norm Alignment

To jusify sampling 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

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

The Taylor expansion of f with $O(\lambda^2)$ terms ignored gives $f(x \pm \lambda u) \approx f(x) \pm \lambda \nabla f(x)^\top u$, hence

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

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.

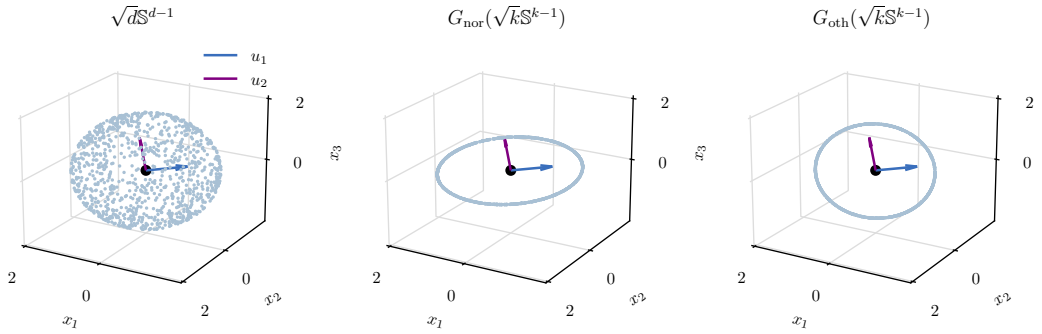

Figure 7: Comparison of the sampled perturbations in full-dimensional space and the public gradient subspace. u_1 and u_2 denote the top-2 left singular vectors of normalized G. *Left*: Vanilla zeroth-order perturbation sampling from $\sqrt{d}\mathbb{S}^{d-1}$. *Middle*: Sampling from $G(\sqrt{k}\mathbb{S}^{k-1})$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

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 sizes, 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. So we have

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

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

Lemma B.2 (Zhang et al. [8], 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 $\sqrt{d}\mathbb{B}^d=\{x\in\mathbb{R}^d\mid\|x\|\le\sqrt{d}\}$. 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 $f(x):\mathbb{R}^d\to\mathbb{R}$ 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*

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

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

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 $f(\cdot)$, running Algorithm 1 for T rounds gives*

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

Additionally, let c_1 and c_2 be the constants that make PAZO-M satisfy (ε,δ)-differential privacy for any $\varepsilon<c_1b^2T/n^2,\delta>0$. Then PAZO-M obtains the error rate

$$O\left(\frac{1-\alpha}{\alpha}\sqrt{d}\right)+O\left(\gamma^2\frac{\alpha\sqrt{d}}{2(1-\alpha)+\alpha\sqrt{d}}+\frac{\sigma_1^2}{b}\frac{(1-\alpha)\sqrt{d}}{(1-\alpha)\sqrt{d}+\alpha}+\frac{\sigma_2^2}{b'}\frac{\alpha^2\sqrt{d}}{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)}\right)$$

by choosing the parameters

$$\eta=\frac{2(1-\alpha)+\alpha\sqrt{d}}{4L((1-\alpha)^2\sqrt{d}+\alpha(1-\alpha))},\quad \lambda\le\frac{1}{Ld^{\frac{5}{4}}},\quad C=1+\sqrt{2}d^{\frac{1}{4}}M,\quad \textit{and}$$

$$T=\frac{4n\varepsilon[(1-\alpha)\sqrt{d}+\alpha]}{c_2C[2(1-\alpha)+\alpha\sqrt{d}]}\sqrt{\frac{2L[f(x_0)-f(x_*)]}{\sqrt{d}\log(1/\delta)}}.$$

Proof. We choose the clipping threshold C large enough such that clipping does not happen, then the update rule is $x_{t+1} - x_t = -\eta_t((1-\alpha)(\Delta(x_t; u_t, B_t) + z_t)u_t + \alpha g'(x_t; B_t'))$ 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., $\mathbb{E}_t \coloneqq \mathbb{E}_{u_t, z_t, B_t, B_t'}$. We have

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

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

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

where T_5 satisfies

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

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

For T_2, note that per-sample L-smoothness implies batch L-smoothness. Therefore, we follow Zhang et al. [8] by noting that

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

where (a) and (b) follow $(a+b)^2 \leq 2(a^2+b^2)$ and (c) follows $|f(x+\lambda u)-f(x)-\lambda u^\top \nabla f(x)| \leq L\lambda^2 d/2$ and $|f(x)-f(x-\lambda u)-\lambda u^\top \nabla f(x)| \leq L\lambda^2 d/2$ due to L-smoothness. Therefore,

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

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

For T_3, applying the equalities

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

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

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

gives us

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

where

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

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

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

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

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

Choosing $\eta_t = \frac{2(1-\alpha)+\alpha\sqrt{d}}{4L((1-\alpha)^2\sqrt{d}+\alpha(1-\alpha))}$, we have $\alpha L\eta_t < 1$ if $\alpha < 1 - \frac{3\sqrt{d}-3}{3\sqrt{d}-2}$. Denote $\mathbb{E}_{<t} \coloneqq \mathbb{E}_{u_{<t},z_{<t},B_{<t},B'_{<t}}$ where $u_{<t}$ is the set $\{u_0,\dots,u_{t-1}\}$ and similarly for $z_{<t}$, $B_{<t}$, and $B'_{<t}$. We sum up from $t=0$ to $T-1$, telescope terms, and divide both sides by T to obtain

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

By privacy analysis in Section 3, we take $\sigma = c_2 b\sqrt{T\log(1/\delta)}/(n\varepsilon)$ and then there exist constants c_1 and c_2 such that PAZO-M is (ε,δ)-differentially private for any $\varepsilon < c_1 b^2T/n^2, \delta > 0$. We apply η_t and σ to Eq. (12) and obtain

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}[\|\nabla f(x_t)\|^2] \\
&\le \frac{16\sqrt{d}L[f(x_0)-f(x_*)]}{T}\frac{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)}{(2(1-\alpha)+\alpha\sqrt{d})^2} + 2L\lambda d^{\frac{5}{4}}M\frac{1-\alpha}{2(1-\alpha)+\alpha\sqrt{d}} \\
&\quad + 2\sqrt{d}\gamma^2\frac{\alpha}{2(1-\alpha)+\alpha\sqrt{d}} + \left[\frac{L^2\lambda^2d^2}{4} + \frac{\sigma_1^2\sqrt{d}}{b} + \frac{c_2^2C^2dT\log(1/\delta)}{2n^2\varepsilon^2}\right]\frac{1-\alpha}{(1-\alpha)\sqrt{d}+\alpha} \\
&\quad + \frac{\sqrt{d}\sigma_2^2}{2b'}\frac{\alpha^2}{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)} + \left[\frac{L\lambda d^{\frac{5}{4}}\gamma}{2} + \left(\gamma + \frac{L\lambda d^{\frac{5}{4}}}{2}\right)M\right]\frac{\alpha}{(1-\alpha)\sqrt{d}+\alpha}. \quad (13)
\end{aligned}$$

To choose the optimal T, we organize the terms involving T, which are of the form $\frac{p}{T} + qT$. We solve $\min_{T>0}\frac{p}{T} + qT = 2\sqrt{pq}$ by taking $T^* = \sqrt{p/q}$, which yields

$$T^* = \frac{4n\varepsilon[(1-\alpha)\sqrt{d}+\alpha]}{c_2C[2(1-\alpha)+\alpha\sqrt{d}]}\sqrt{\frac{2L[f(x_0)-f(x_*)]}{\sqrt{d}\log(1/\delta)}}.$$

By $\Delta(x_t;u_t,\xi_i)^2 \le \frac{L^2\lambda^2d}{2} + 2(u_t^\top\nabla f(x_t;\xi_i))^2$ and per-sample M-Lipschitz, we have

$$\Delta(x_t;u_t,\xi_i) \le \sqrt{d^{-\frac{3}{2}}/2 + 2\sqrt{d}M^2} \le 1 + \sqrt{2}d^{\frac{1}{4}}M$$

due to $\sqrt{p+q} \leq \sqrt{p} + \sqrt{q}$ for $p, q \geq 0$. We choose $C = 1 + \sqrt{2}d^{\frac{1}{4}}M$ and $\lambda \leq \frac{1}{Ld^{\frac{5}{4}}}$ and thus have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}[\|\nabla f(x_t)\|^2] \\
\leq\ & \frac{4c_2(1+\sqrt{2}d^{\frac{1}{4}}M)(1-\alpha)d^{\frac{3}{4}}}{n\varepsilon[2(1-\alpha)+\alpha\sqrt{d}]}\sqrt{2L[f(x_0)-f(x_*)]\log(1/\delta)} + 2M\frac{1-\alpha}{2(1-\alpha)+\alpha\sqrt{d}} \\
&+ 2\gamma^2\frac{\alpha\sqrt{d}}{2(1-\alpha)+\alpha\sqrt{d}} + \left[\frac{1}{4\sqrt{d}} + \frac{\sigma_1^2\sqrt{d}}{b}\right]\frac{1-\alpha}{(1-\alpha)\sqrt{d}+\alpha} \\
&+ \frac{\sigma_2^2}{2b'}\frac{\alpha^2\sqrt{d}}{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)} + \left[\frac{\gamma}{2} + \left(\gamma+\frac{1}{2}\right)M\right]\frac{\alpha}{(1-\alpha)\sqrt{d}+\alpha},
\end{aligned}
$$

which indicates that the error depends on d, σ_1, σ_2, and γ by

$$
O\left(\frac{1-\alpha}{\alpha}\sqrt{d}\right) + O\left(\gamma^2\frac{\alpha\sqrt{d}}{2(1-\alpha)+\alpha\sqrt{d}} + \frac{\sigma_1^2}{b}\frac{(1-\alpha)\sqrt{d}}{(1-\alpha)\sqrt{d}+\alpha} + \frac{\sigma_2^2}{b'}\frac{\alpha^2\sqrt{d}}{(1-\alpha)^2\sqrt{d}+\alpha(1-\alpha)}\right).
$$

Therefore, we have error dependence $O(\frac{1-\alpha}{\alpha}\sqrt{d})$, which saves a factor of $\log d$ compared to DPZero's $O(\sqrt{d}\log d)$, together with constant improvement if $\alpha > \frac{1}{2}$. We additionally have the error term $O(\gamma^2 + \sigma_2^2/b')$ that reduces as α decreases due to using biased public gradients.

□

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 $f(\cdot)$, running Algorithm 2 for T rounds gives*

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

Additionally, let c_1 and c_2 be the constants that make PAZO-M satisfy (ε, δ)-differential privacy for any $\varepsilon < c_1 b^2 T/n^2, \delta > 0$. Then PAZO-P obtains the error rate

$$
O(k) + O\left(\sqrt{\gamma^2 + \frac{\sigma_2^2}{b'}} + \frac{\sigma_1^2}{b}\right)
$$

by choosing the parameters

$$
\eta = \frac{1}{2Lk}, \quad \lambda \leq \frac{1}{Lk^{\frac{3}{2}}}, \quad C = 1 + \sqrt{2k}M, \quad \text{and } T = \frac{n\varepsilon}{c_2 C}\sqrt{\frac{8Lk[f(x_0)-f(x_*)]}{\log(1/\delta)}}.
$$

Proof. We choose the clipping threshold C large enough such that clipping does not happen, then the update rule is $x_{t+1} - x_t = -\eta_t(\Delta(x_t; u_t, B_t) + z_t)G_t u_t$ where

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

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., $\mathbb{E}_t \coloneqq \mathbb{E}_{u_t, z_t, B_t, B_t'}$. We have

$$
\begin{aligned}
&\mathbb{E}_t[f(x_{t+1})] \\
\leq & f(x_t) + \langle \nabla f(x_t), \mathbb{E}_t[x_{t+1} - x_t]\rangle + \frac{L}{2}\mathbb{E}_t[\|x_{t+1} - x_t\|^2]
\end{aligned}
$$

$$
\begin{aligned}
&= f(x_t) - \eta_t \langle \nabla f(x_t), \mathbb{E}_t[\Delta(x_t; u_t, B_t) G_t u_t] \rangle + \frac{L\eta_t^2}{2} \mathbb{E}_t[\|\Delta(x_t; u_t, B_t) G_t u_t\|^2] + \frac{L\eta_t^2}{2} \mathbb{E}_t \left[\left\| \frac{z_t}{b} G_t u_t \right\|^2 \right] \\
&\overset{(a)}{=} f(x_t) - \eta_t \|\nabla f(x_t)\|^2 + \eta_t \underbrace{\langle \nabla f(x_t), \nabla f(x_t) - \mathbb{E}_t[\Delta(x_t; u_t, B_t) G_t u_t] \rangle}_{T_1} \\
&\quad + \frac{L\eta_t^2 k}{2} \underbrace{\mathbb{E}_t[\|\Delta(x_t; u_t, B_t)\|^2]}_{T_2} + \frac{L\eta_t^2 \sigma^2 C^2 k}{2b^2}, \qquad (14)
\end{aligned}
$$

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

For T_1, we proceed by

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

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

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

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

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

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

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

For T_3, we thus have

$$
\begin{aligned}
\left\| \nabla f(x_t) - \mathbb{E}_t[G_t G_t^\top \nabla f(x_t)] \right\| &\leq \mathbb{E}_t[\left\| \nabla f(x_t) - G_t G_t^\top \nabla f(x_t) \right\|] \\
&\leq \sqrt{2(\sigma_2^2/b' + \gamma^2)}. \qquad (15)
\end{aligned}
$$

For T_4, we have

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

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

$$T_1 \leq M\left(\sqrt{2(\sigma_2^2/b' + \gamma^2)} + \frac{L\lambda k^{\frac{3}{2}}}{2}\right). \tag{16}$$

For T_2, note that

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

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

$$\begin{aligned}
\mathbb{E}_t[\|\Delta(x_t; u_t, B_t)\|^2] &= \frac{L^2\lambda^2 k^2}{2} + 2\mathbb{E}_{B_t, B_t'}\mathbb{E}_{u_t}[(u_t^\top G_t^\top \nabla f(x_t; B_t))^2] \\
&= \frac{L^2\lambda^2 k^2}{2} + 2\mathbb{E}_{B_t, B_t'}[\left\|G_t^\top \nabla f(x_t; B_t)\right\|^2] \\
&= \frac{L^2\lambda^2 k^2}{2} + 2\mathbb{E}_{B_t, B_t'}[\nabla f(x_t; B_t)^\top G_t G_t^\top \nabla f(x_t; B_t)] \\
&= \frac{L^2\lambda^2 k^2}{2} + 2\mathbb{E}_{B_t, B_t'}[\nabla f(x_t; B_t)^\top \text{Proj}_G(\nabla f(x_t; B_t))] \\
&\leq \frac{L^2\lambda^2 k^2}{2} + 2\mathbb{E}_{B_t}[\|\nabla f(x_t; B_t)\|^2] \\
&\leq \frac{L^2\lambda^2 k^2}{2} + 2\left(\|\nabla f(x_t)\|^2 + \frac{\sigma_1^2}{b}\right).
\end{aligned} \tag{17}$$

Applying T_1 (16) and T_2 (17) to (14) yields

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

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

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

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

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{<t}[\|\nabla f(x_t)\|^2] \leq \frac{4Lk}{T}[f(x_0) - f(x_*)] + 2M\sqrt{2\left(\frac{\sigma_2^2}{b'} + \gamma^2\right)} \\
+ L\lambda k^{\frac{3}{2}} M + \frac{L^2\lambda^2 k^2}{4} + \frac{\sigma_1^2}{b} + \frac{\sigma^2 C^2}{2b^2}.
\end{aligned} \tag{18}$$

By privacy analysis in Section 3, we take $\sigma = c_2 b\sqrt{T\log(1/\delta)}/(n\varepsilon)$ and then there exist constants c_1 and c_2 such that PAZO-P is (ε, δ)-differentially private for any $\varepsilon < c_1 b^2 T/n^2, \delta > 0$. We apply η_t and σ to Eq. (18) and obtain the RHS of Eq. (18) as

$$\frac{4Lk[f(x_0) - f(x_*)]}{T} + 2M\sqrt{2\left(\frac{\sigma_2^2}{b'} + \gamma^2\right)} + L\lambda k^{\frac{3}{2}} M + \frac{L^2\lambda^2 k^2}{4} + \frac{\sigma_1^2}{b} + \frac{c_2^2 C^2 \log(1/\delta) T}{2n^2\varepsilon^2}.$$

Choosing the optimal T again requires solving $\arg\min_{T>0} \frac{p}{T} + qT = \sqrt{p/q}$, which yields

$$T^* = \frac{n\varepsilon}{c_2 C}\sqrt{\frac{8Lk[f(x_0) - f(x_*)]}{\log(1/\delta)}}.$$

By $\Delta(x_t; u_t, \xi_i)^2 \leq \frac{L^2\lambda^2 k^2}{2} + 2(u_t^\top G_t^\top \nabla f(x_t; \xi_i))^2$ and per-sample M-Lipschitz, we have

$$\Delta(x_t; u_t, \xi_i) \leq \sqrt{k^{-1}/2 + 2kM^2} \leq 1 + \sqrt{2k}M.$$

We take $C = 1 + \sqrt{2k}M$ and choose $\lambda \leq \frac{1}{Lk^{\frac{3}{2}}}$, and thus we have the RHS of Eq. (18) as

$$\frac{2(1+\sqrt{2k}M)c_2}{n\varepsilon}\sqrt{2Lk[f(x_0) - f(x_*)]\log(1/\delta)} + 2M]\sqrt{2\left(\frac{\sigma_2^2}{b'} + \gamma^2\right)} + M + \frac{1}{4k} + \frac{\sigma_1^2}{b},$$

which indicates that the error depends on k, σ_1, σ_2, and γ by

$$O(k) + O\left(\sqrt{\frac{\sigma_2^2}{b'} + \gamma^2} + \frac{\sigma_1^2}{b}\right).$$

Therefore, we have d-independent error rate $O(k)$, which is an improvement due to k being a small constant $\ll \log d$ in practice. We additionally have the error term $O(\gamma^2 + \sigma_2^2/b')$ from the biased public gradients and $O(\sigma_1^2/b)$ from the stochastic private gradients. □

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 $f(\cdot)$, running Algorithm 3 for T rounds using a fixed step size $\eta = \frac{1}{4L}$ and $\epsilon \leq 1/\sqrt{d}$ gives*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{<t}[\|\nabla f(x_t)\|^2] \leq \frac{8L\mathbb{E}_{<t+1}[f(x_0) - f(x_*)]}{T} + 2M\left(\gamma + \frac{\sigma_2}{\sqrt{b'}}\right) + 2\gamma^2 + \frac{2\sigma_2^2}{b'} + \frac{1}{2}.$$

Additionally, let c_1 and c_2 be the constants that make PAZO-S satisfy (ε, δ)-differential privacy for any $\varepsilon < c_1 b^2 T/n^2, \delta > 0$. Then by taking $T \to \infty$, PAZO-S obtains the error rate $O\left(\gamma^2 + \sigma_2^2/b'\right)$.

Proof. Our public data sampling process is equivalent to first sampling B'_t and then dividing it into k non-overlapping partitions. We choose the clipping threshold C large enough such that clipping does not happen, then the update rule is $x_{t+1} - x_t = -\eta_t(g'(x_t; B'_{t,I}) + \mathbb{1}(z')z')$ where $I \coloneqq \arg\min_{i\in[k]}\{f(x_t - \eta_t g(x_t; B'_{t,i}); B_t) + z_{t,i}\}$ is the index of public batch that yields the best public gradients among and $\mathbb{1}(z')$ is an indicator variable denoting whether the proposal of adding $z' \sim \mathcal{N}(0, \epsilon^2 I_d)$ is adopted.

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., $\mathbb{E}_t \coloneqq \mathbb{E}_{z_t, B_t, B'_t}$.

$$
\begin{aligned}
&\mathbb{E}_t[f(x_{t+1})] \\
&= \mathbb{E}_t[f(x_t - \eta_t(g'(x_t; B'_{t,I}) + \mathbb{1}(z')z'))] \\
&\leq f(x_t) - \eta_t \left\langle \nabla f(x_t), \mathbb{E}_t[g'(x_t; B'_{t,I}) + \mathbb{1}(z')z'] \right\rangle + \frac{L\eta_t^2}{2} \underbrace{\mathbb{E}_t[\|g'(x_t; B'_{t,I}) + \mathbb{1}(z')z'\|^2]}_{T_1} \\
&= f(x_t) - \eta_t \left\langle \nabla f(x_t), \mathbb{E}_t[g'(x_t; B'_{t,I})] \right\rangle + \frac{L\eta_t^2}{2} T_1 \\
&= f(x_t) - \eta_t \|\nabla f(x_t)\|^2 + \eta_t \left\langle \nabla f(x_t), \nabla f(x_t) - \mathbb{E}_t[g'(x_t; B'_{t,I})] \right\rangle + \frac{L\eta_t^2}{2} T_1 \\
&\leq f(x_t) - \eta_t \|\nabla f(x_t)\|^2 + \eta_t \|\nabla f(x_t)\| \underbrace{\mathbb{E}_t[\|\nabla f(x_t) - g'(x_t; B'_{t,I})\|]}_{T_2} + \frac{L\eta_t^2}{2} T_1.
\end{aligned}
$$

For T_1, we have

$$
\begin{aligned}
\mathbb{E}_t[\|g'(x_t; B'_{t,I}) + \mathbb{1}(z')z'\|^2] &\leq 2\mathbb{E}_t[\|g'(x_t; B'_{t,I})\|^2] + 2\mathbb{E}_t[\|\mathbb{1}(z')z'\|^2] \\
&\leq 2\mathbb{E}_t[\|g'(x_t; B'_{t,I}) - \nabla f(x_t) + \nabla f(x_t)\|^2] + 2d\epsilon^2 \\
&\leq 4\mathbb{E}_t[\|g'(x_t; B'_{t,I}) - \nabla f(x_t)\|^2] + 4\|\nabla f(x_t)\|^2 + 2d\epsilon^2 \\
&= 4\mathbb{E}_t[\|g'_t - g_t + \frac{1}{b'} \sum_{j \in B'_t} \zeta_{t,j}^{(I)'} \|^2] + 4\|\nabla f(x_t)\|^2 + 2d\epsilon^2 \\
&\leq 8\gamma^2 + \frac{8\sigma_2^2}{b'} + 4\|\nabla f(x_t)\|^2 + 2d\epsilon^2.
\end{aligned}
$$

For T_2, we note that for a sampled public batch $i \in [k]$, its gradient is $g'(x_t; B'_{t,i}) = g'_t + \frac{1}{b'} \sum_{j=1}^{b'} \zeta_{t,j}^{(i)'}$ where $\zeta_{t,j}^{(i)'}$ is the stochastic gradient noise for the public sample j in the i-th batch. We denote the selected best batch as I and thus

$$
\mathbb{E}_{B'_t}[\left\|g'_t - g'(x_t; B'_{t,I})\right\|^2] = \mathbb{E}_{B'_t}\left[\left\|\frac{1}{b'} \sum_{j=1}^{b'} \zeta_{t,j}^{(I)'}\right\|^2\right] = \frac{1}{b'} \mathbb{E}_{B'_t}\left[\left\|\zeta_t^{(I)'}\right\|^2\right].
$$

By assumption, $\mathbb{E}_{B'_t}\left[\left\|\zeta_t^{(i)'}\right\|^2\right] \leq \sigma_2^2$ for any batch i. Therefore,

$$
\mathbb{E}_{B'_t}\left[\left\|\zeta_t^{(I)'}\right\|^2\right] = \mathbb{E}_i\left[\mathbb{E}_{B'_t}\left[\left\|\zeta_t^{(I)'}\right\|^2\right] | I = i\right] \leq \sigma_2^2.
$$

Therefore, $(\mathbb{E}_t[\left\|g_t - g'(x_t; B'_{t,I})\right\|])^2 \leq \mathbb{E}_t[\left\|g_t - g'(x_t; B'_{t,I})\right\|^2] \leq \sigma_2^2/b'$ and

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

Denote $\mathbb{E}_{<t} \coloneqq \mathbb{E}_{z_{<t}, B_{<t}, B'_{<t}}$ where $z_{<t}$ is the set $\{z_0, \ldots, z_{t-1}\}$ and similarly for $B_{<t}$ and $B'_{<t}$. We have

$$
(\eta_t - 2L\eta_t^2)\mathbb{E}_{<t}\|\nabla f(x_t)\|^2 \leq \mathbb{E}_{<t+1}[f(x_t) - f(x_{t+1})] + \eta_t M\left(\gamma + \frac{\sigma_2}{\sqrt{b'}}\right) + 4L\eta_t^2\left(\gamma^2 + \frac{\sigma_2^2}{b'} + \frac{d\epsilon^2}{4}\right).
$$

We set $\epsilon \leq 1/\sqrt{d}$ and choose $\eta_t = \frac{1}{4L}$ so that $2L\eta_t^2 = \eta_t/2$. We sum up from $t = 0$ to $T - 1$, and dividing both sides by T yields

$$
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}_{<t}[\|\nabla f(x_t)\|^2] \leq \frac{8L\mathbb{E}_{<t+1}[f(x_0) - f(x_*)]}{T} + 2M\left(\gamma + \frac{\sigma_2}{\sqrt{b'}}\right) + 2\gamma^2 + \frac{2\sigma_2^2}{b'} + \frac{1}{2}.
$$

We take $T \to \infty$ and achieve a d-independent error bound $O(\gamma^2 + \sigma_2^2/b')$. When γ approaches zero, the remaining term σ_2^2/b' is due to stochastic public data sampling. $\square$

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 [41] 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 [43].

Tiny-ImageNet. We follow Kurakin et al. [44], which first pre-trains a ResNet18 on Places365 [45] 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) [46] with random initialization.

IMDB. We follow Li et al. [16], which uses Amazon Polarity [47] 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 [6, 8] 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 [48] 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 ($\varepsilon, \delta = 1/\#\text{train samples}$) 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 [40]), we adopt their provided implementation and privacy accounting.

The experiment on MNLI utilizes the codebase from Malladi et al. [6] and Zhang et al. [8], 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 [8] 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 $0.1\% \sim 2.5\%$.

Performance of public only. We demonstrate that the improvements of using public data are not due to overfitting to public data. We train on public data alone using SGD with batch size, learning rate, and weight-decay tuned, and the optimal hyperparameter for each setting gives us accuracies equal to 66.1% for CIFAR-10, 27.1% for Tiny-ImageNet, 68.4% for IMDB, and 60.8% for MNLI. We denote these results as 'public only' and pick the best first-order with public data (FO+PUB) and zeroth-order with public data (PAZO) algorithm for each dataset. In Table 5, we present the performance gain when private data is included (i.e., 'FO+PUB/PAZO' minus 'public only' across $\varepsilon = 0.1, 0.5, 1, 2, 3$). Note that 'public only' accuracies come with severe overfitting due to the small number of public samples, while the DP accuracies are not overfit.

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

Type	Method	$\varepsilon = 0.1$	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$	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	<u>70.9</u>	**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**	<u>71.2</u>	<u>70.5</u>	
	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	<u>70.3</u>	70.3	70.2	69.8	69.7	

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

Type	Method	$\varepsilon = 0.1$	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	Non-private
FO	DP-SGD	24.8	29.2	31.4	38.0	52.9
FO+PUB	DPMD	30.5	<u>31.4</u>	**34.2**	**35.5**	
	DOPE-SGD	30.7	**31.8**	<u>32.5</u>	<u>34.4</u>	
	GEP	–	30.9	30.5	31.4	
ZO	DPZero	25.1	27.6	27.5	27.9	28.6
ZO+PUB (ours)	PAZO-M	<u>30.8</u>	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	$\varepsilon = 0.1$	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$	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	<u>76.6</u>	76.6	
	DOPE-SGD	70.2	73.2	**75.0**	75.9	<u>77.9</u>	
	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	<u>73.4</u>	73.2	<u>74.5</u>	73.2	73.6	
	PAZO-P	71.0	<u>73.7</u>	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	$\varepsilon = 0.1$	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$	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	<u>70.1</u>	<u>72.5</u>	
	GEP	–	–	–	–	–	
ZO	DPZero	–	55.2	58.2	60.4	62.6	68.4
ZO+PUB (ours)	PAZO-M	<u>67.1</u>	67.3	67.8	67.7	67.5	
	PAZO-P	63.5	<u>68.3</u>	**69.8**	69.7	70.3	
	PAZO-P′	61.0	68.1	68.8	69.0	69.4	
	PAZO-S	**68.2**	**68.6**	<u>68.9</u>	68.6	69.0	

Table 5: Performance of training with public data only and the improvements from using private data via first-order (FO+PUB) and zeroth-order (PAZO) methods. We observe that (1) FO enjoys up to 12.0% performance gain and ZO enjoys up to 8.2% when private data is included; (2) ZO consistently enjoys performance gain when private data is included, while FO does not, since private first-order gradients can be too noisy under tight privacy.

	CIFAR-10	Tiny-ImageNet	IMDB	MNLI
Public only	66.1	27.1	68.4	60.8
FO+PUB improvement	$-1.3 \sim 6.8$	$3.6 \sim 8.4$	$2.6 \sim 10.2$	$-1.1 \sim 12.0$
PAZO improvement	$4.4 \sim 5.2$	$3.8 \sim 4.1$	$5.3 \sim 6.2$	$7.4 \sim 9.5$

Table 6: Performance under different public data with γ under privacy $\varepsilon = 1.0$. We observe that though the range of γ-similarity depends on specific methods, the values are consistently bounded and small. For a fixed PAZO variant, as the public data becomes more in-distribution, performance improves and γ decreases (i.e., gradients for public and private data become more similar).

Private Data	Public Data	PAZO-M	PAZO-P	PAZO-S
CIFAR-10	Slight imbalance	71.3 ($\gamma = 4.3$)	69.0 ($\gamma = 3.4$)	70.2 ($\gamma = 1.6$)
	Half-half	69.9 ($\gamma = 4.8$)	67.3 ($\gamma = 4.1$)	67.3 ($\gamma = 1.8$)
	Big imbalance	66.7 ($\gamma = 6.2$)	63.8 ($\gamma = 4.8$)	63.0 ($\gamma = 2.1$)
MNLI	MNLI only	75.7 ($\gamma = 39$)	74.1 ($\gamma = 41$)	74.0 ($\gamma = 49$)
	Half-half	73.2 ($\gamma = 50$)	73.8 ($\gamma = 43$)	73.1 ($\gamma = 67$)
	SNLI only	67.8 ($\gamma = 71$)	69.8 ($\gamma = 65$)	68.9 ($\gamma = 81$)

Performance under various γ. We demonstrate that PAZO performs better when public data is closer to the private data in two settings: pre-training on CIFAR-10 and fine-tuning with prompts on MNLI. To create public data of different extents of distribution shifts (different γ's), we mix ID public data and OOD public data with different proportions. For CIFAR-10, we use non-overlapped training samples with small class imbalance as ID public data and those with big class imbalance as OOD public data. The slight class imbalance has class-size ratios $[1 : \ldots : 0.85]$ and big class imbalance has class-size ratios $[1.0 : 0.9 : 0.8 : \ldots : 0.2 : 0.1]$. For MNLI, we use non-overlapped MNLI training samples as ID public data and SNLI training samples as OOD public data. We present the performance and γ of PAZO under these scenarios in Table 6. We observe that (1) the range of γ is method-dependent and (2) for any fixed PAZO variant, the accuracy increases as the data become more similar (smaller γ's).

Runetime efficiency. Theoretically, we list the number of different types of operations involved in each algorithm in Table 8. Since the first-order methods require per-sample gradient computation and clipping, the number of "gradient backward", the slowest operation, is dependent on the private batch size. This is a discouraging feature since large batch sizes offer better utility/privacy tradeoffs [14, 42], 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 7). We vectorize the three settings other than the IMDB-LSTM experiment due to incompatibility between the model architecture and `vmap`. Although the MNLI experiments enjoys only $2\times$ of speedup by using PAZO, Malladi et al. [6] shows that zeroth-order methods will be significantly faster as the model scales up.

Memory efficiency. Table 8 presents the number of different operations needed per iteration of each method, showing that PAZO-{M,P,S} has memory overhead to store public gradients compared to DPZero. PAZO-M requires one batch of public gradient, so the memory overhead is $O(d)$, where d is the number of model parameters. PAZO-S is also $O(d)$ since we can compute the k public batch gradients sequentially. Though PAZO-P has an $O(kd)$ memory overhead than DPZero, it is still more memory- and computation-efficient than the first-order DP methods since the latter generally

Table 7: Speed of each method on different datasets (in s/iter). It shows that PAZO offers up to $16\times$ runtime speedup per training iteration compared to the baselines. All numbers are averaged over 20 iterations. Note that we report the speed of each method under optimal (k, b', q). DPZero is occasionally slower than PAZO because we try $q = \{1, 5\}$ for each method and observe that DPZero needs $q = 5$ while PAZO can take $q = 1$ to achieve competitive accuracies.

	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×

Table 8: 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	–

requires $O(bd)$ memory to maintain per-sample gradients. Our experimental results are obtained using $k = \{3, 6\}$ while $b = 64$. Such entangled dependence on b and d is also restrictive since larger batch sizes improve performance [14, 49].

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 [8, 6], 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 9. We also report the hyperparameter search grid for each method in Table $11-12$, where the batch size b is only tuned for non-private methods (SGD and MeZO); We fix the private batch size to 64 for all private methods, including zeroth-order and first-order, with and without public data.

Sensitivity to q. Table 10 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.

Table 9: 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 10: 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 in accuracies by 1.0%, 2.4%, 4.8%, and 7.2% 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

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.

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.

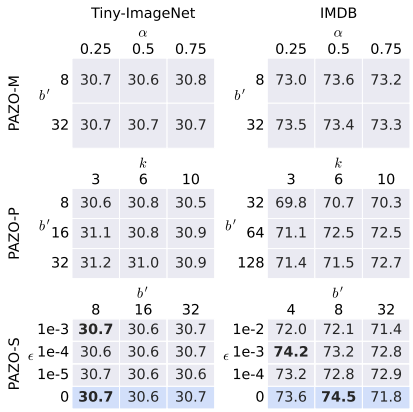

Figure 8: All PAZO methods are robust to different values of their introduced hyperparameters. Each number represents the best accuracy with standard hyperparameters for zeroth-order private methods (C and η) tuned. Blue cells indicate PAZO-S performance without having a noisy candidate.

Table 11: 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	η	{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 12: 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	η	{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}

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We summarize the contributions in the abstract and introduction, with our proposed algorithm described in Section 3, theoretical analyses made in Section 4, and experiments detailed in Section 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We list the assumptions and summarize the results in Section 4. We provide the complete proofs in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the experimental setup in Section 5 and hyperparameter tuning in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Datasets are publicly accessible and code is released on GitHub.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the algorithm details in Section 3, experimental setup in Section 5, the full experiment results in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper does not report error bars in experimental evaluation. However, all the experiments use the random seed 0, so the results are not biased. Importantly, we report the results under multiple hyperparameters in Section 5 and Appendix C.3. Results show that the performance of our methods is consistent across different hyperparameter values, which illustrates the statistical significance of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss the compute type we use and the runtime of both our methods and the baselines in Section 5 and Appendix C.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed Code and Ethics, and confirm that our submission conforms with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader social impacts in abstract and Section 1.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited the data and models we use in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We use open-sourced data and models in our work and they are properly referred.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This paper uses LLM only for grammar editing and formatting improvement, so declaration is not made.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.