Private Fine-tuning of Large Language Models with Zeroth-order Optimization

Anonymous Authors[1]

Abstract

Differential privacy is a framework for mitigating privacy risks by enforcing algorithmic stability. DP-SGD allows models to be trained in a privacy-preserving manner, but raises new obstacles in the form of performance loss and significant engineering challenges. We introduce DP-ZO, a new method for fine-tuning large language models that preserves the privacy of training data by privatizing zeroth-order optimization. A key insight into the design of our method is that the direction of the gradient in the zeroth-order optimization we use is random and the only information from training data is the step size, i.e., a scalar. Therefore, we only need to privatize the scalar step size, which is memory-efficient. DP-ZO, which can be instantiated with either Laplace or Gaussian noise, provides a strong privacy-utility trade-off across different tasks, and model sizes, under conservative privacy budgets.

1. INTRODUCTION

The proliferation of open-source models pretrained on web-scale datasets (Brown et al., 2020; Zhang et al., 2022; Touvron et al., 2023) has created a paradigm shift in privacy preserving machine learning. Differential Privacy (DP) (Dwork et al., 2006) is the gold standard for preserving privacy while training models on private data, but it requires additional data (Tramèr and Boneh, 2021) to prevent a drop in utility (Yu et al., 2021a). Pretrained model checkpoints have emerged as a compelling "free" source of prior information to boost the performance of DP training (Ganesh et al., 2023; Tang et al., 2023a; Panda et al., 2022). By only requiring DP during the fine-tuning phase, a recent line of work (Li et al., 2022b;a; Yu et al., 2021b; He et al., 2023; Bu et al., 2023c) is able to obtain impressive performance with DP-SGD (Abadi et al., 2016). Despite these advancements, DP-SGD needs additional engineering effort, especially for large models across devices. We propose a new direction for DP fine-tuning of large pretrained models that achieves strong privacy-utility trade-off and is more resource-efficient, easy to implement, and portable.

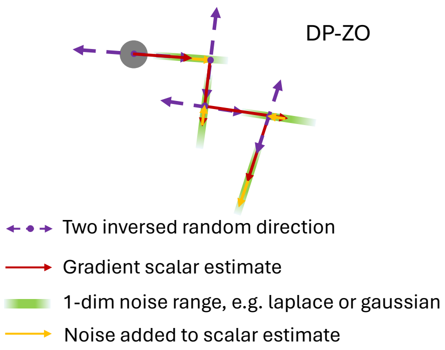
DP-ZO
Two inversed random direction
Gradient scalar estimate
1-dim noise range, e.g. laplace or gaussian
Noise added to scalar estimate

Figure 1. Visualization of DP-ZO. The only information from private data is a scalar and we only need to add noise to this scalar. This scalar privatization enjoys the benefits of flexibility with DP mechanisms, ease of implementation, and reduced computation.

In this work, we introduce a *new methodology DP-ZO* for DP fine-tuning of large pretrained models. Our method uses zeroth-order optimization (ZO) (Spall, 1992). Our key insight is the synergy between differentially private fine-tuning and zeroth-order optimization. ZO provides the gradient estimates and the only information from private data in ZO is a scalar. We only need to privatize the scalar update by adding noise to it. Specifically, the scalar is the differences between losses from models with the same random perturbation but flipped signs. DP-ZO privatizes the zeroth-order update, by adding noise to the difference between the losses (visualized in Figure 1). This noise is proportional to the sensitivity of this loss difference with respect to changing a single example in the training set, which is controlled by clipping. We limit the ℓ_p sensitivity by clipping the norm of the difference in scalar losses, between the two random perturbations. Therefore, DP-ZO is flexible for both for ε-DP and (ε, δ)-DP. By removing the need for per-example gradient clipping (Abadi et al., 2016),

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2024 Workshop on Foundation Models in the Wild. Do not distribute.

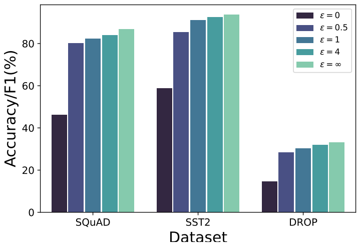

Figure 2. DP-ZO provides a strong privacy-utility trade-off across different tasks under conservative privacy budgets.

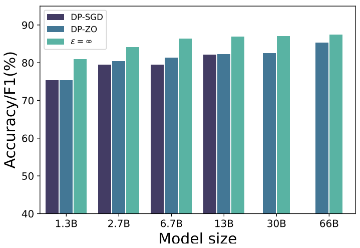

Figure 3. DP-ZO achieves comparable performance as DP-SGD with same model size and scales seamlessly to large models like 30B/66B, that are challenging for DP-SGD.

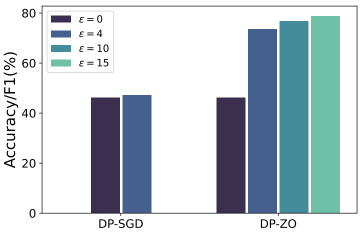

Figure 4. DP-ZO achieves non-trivial performance for ε-DP. In contrast, DP-SGD (laplace) suffers to improve upon $\varepsilon = 0$ due to high variance.

DP-ZO enables DP training of language models with just a few lines of code and without the need for backpropagation.

Main results. We presents the main results of DP-ZO in Figures 2 to 4. DP-ZO provides a strong privacy-utility trade-off across different tasks, model sizes, dataset sizes, and DP mechanisms under conservative privacy budgets. DP-ZO only slightly degrades the performance compared to the non-private baseline (Figure 2). DP-ZO achieves comparable performance as DP-SGD within the same model size from 1.3B to 13B (Figure 3). DP-ZO scales seamlessly to large models without additional engineering, while DP-SGD requires much more memory and effort to implement per-example gradient clipping across GPUs (within a reasonable research computation limit, DP-SGD results on OPT-30B/66B are not available and omitted in Figure 3). As the model size increases to OPT-66B, the performance of DP-ZO increases and the utility gap between DP-ZO and the non-private baseline also decreases (Figure 3). Because our method only privatizes a scalar, it is compatible with multiple DP mechanisms. Specifically, DP-ZO is the first method to provide pure ε-DP with nontrivial utility (73.52 for SQuAD at $\varepsilon = 4$) for large models by using the Laplace mechanism (Figure 4).

2. BACKGROUND

2.1. Differential Privacy

Differential privacy (DP) is the gold standard method for providing algorithmic privacy (Dwork et al., 2006).

Definition 2.1 ((ε, δ)$-$ Differential Privacy (DP))**.** An algorithm $\mathcal{M}$ is said to be (ε, δ)-DP if for all sets of events $S \subseteq \text{Range}(\mathcal{M})$ and neighboring datasets $D, D' \in \mathcal{D}^n$ (where $\mathcal{D}$ is the set of all possible data points) we have the guarantee:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in S] + \delta \qquad (1)$$

When $\delta = 0$, we term it as pure $(\varepsilon, 0)$-DP or ε-DP for simplicity.

We define a set of existing DP mechanisms that we will use in our work.

Proposition 2.2 (Gaussian mechanism (Dwork and Roth, 2014))**.** *For any function $f : \mathbb{X}^n \to \mathbb{R}$ with l_2 sensitivity Δ, the mechanism defined as*

$$M(X) = f(X) + z,$$

where $z \sim N\left(0, \frac{2\ln(1.25/\delta)\Delta^2}{\varepsilon^2}\right)$, provides (ε, δ)-DP.

Proposition 2.3 (Laplace mechanism (Dwork and Roth, 2014))**.** *For any function $f : \mathbb{X}^n \to \mathbb{R}$ with l_1 sensitivity Δ the mechanism defined as*

$$M(X) = f(X) + z,$$

where $z \sim$ Laplace $\left(0, \frac{\Delta}{\varepsilon}\right)$, provides $(\varepsilon, 0)$-DP.

2.2. Zeroth-order Optimization

We use a method from the large body of work on zeroth-order optimization (Flaxman et al., 2004; Shamir, 2013; Ghadimi and Lan, 2013; Nesterov and Spokoiny, 2017) that uses the difference in losses between two random perturbations (Duchi et al., 2015; Spall, 1992) with opposite signs to determine the magnitude of a gradient update in the direction of the random perturbations. In the non-private setting where the adaptation between the pretrained model and the fine-tuning dataset has low rank (Hu et al., 2022), as in fine-tuning large language models, Malladi et al. (2023) show this method converges at a rate that is not catastrophically slower than SGD fine-tuning.

Definition 2.4 (Simultaneous Perturbation Stochastic Approximation (SPSA) (Spall, 1992)))**.** Given a model with parameters $\theta \in \mathbb{R}^d$ and a loss function $\mathcal{L}$, the gradient estimate on a minibatch drawn from a dataset $\mathcal{B} \subset \mathcal{D}$ is computed by projecting the loss on the minibatch $\mathcal{L}(\theta; \mathcal{B})$ onto a random perturbation $z \in \mathbb{R}^d$ that is a standard Gaussian random variable (i.e., $z \sim \mathcal{N}(0, I_d)$) scaled by ϕ:

$$\nabla \hat{\mathcal{L}}_b(\theta; \mathcal{B}) = \frac{\mathcal{L}(\theta + \phi z; \mathcal{B}) - \mathcal{L}(\theta - \phi z; \mathcal{B})}{2\phi} z \qquad (2)$$

Random perturbations in zeroth-order optimization (ZO) serve as high-variance estimates of the actual gradient, enabling optimization without the need for explicit gradient computations. However, these perturbations themselves carry a privacy risk. The characteristics of the perturbations can be inferred from the gradient updates, effectively leaking information about the data. Moreover, the magnitude of the perturbations can be estimated with high precision in high dimensions due to the central limit theorem. Therefore, incorporating differential privacy into ZO is essential to safeguard against these vulnerabilities.

3. OUR METHOD: DP-ZO

We introduce our framework for differentially private zeroth order optimization (DP-ZO) by integrating DP into Definition 2.4. In our framework, the information obtained from training data can be represented as a scalar. This scalar has a bounded sensitivity (when applying clipping) and can be privatized by adding noise. If we compare the noise added in DP-ZO to a single dimension to the noise added in DP-SGD to the entire gradient, we expect the univariate noise to be less detrimental to the utility (due to the curse dimensionality in differential privacy (Dwork and Roth, 2014)). In other words, we would expect the gap between non-private and private utility to be smaller than that of DP-SGD.

Algorithm 1 Differentially Private-ZO

```
Model parameters $\theta$, dataset $\mathcal{D}$, learning rate $\eta$, perturbation scale $\phi$, privacy parameter $\sigma$, noising mechanism $\mathcal{Z}$, clipping threshold $C$, expected batch size $B$, sub-sampling rate $p = B/|\mathcal{D}|$).
$g = 0$
for $t \in 1, \dots T$ do
    Poisson sample $\mathcal{B}$ from $\mathcal{D}$ with sub-sampling rate $p$
    $\vec{z} \sim \mathcal{N}(\vec{0}_{|\theta|}, \mathbf{I}_{|\theta| \times |\theta|})$
    $\theta^{+} \leftarrow \theta + \phi\vec{z}$
    $\theta^{-} \leftarrow \theta - \phi\vec{z}$
    for $(x_i, y_i) \in \mathcal{B}$ do
        $l_i^{+} \leftarrow \mathcal{L}(\theta^{+}, (x_i, y_i))$
        $l_i^{-} \leftarrow \mathcal{L}(\theta^{-}, (x_i, y_i))$
        $l_i = clip(l_i^{+} - l_i^{-}, C)$
    end for
    $s = \frac{\sum_{i \in \mathcal{B}} l_i + \mathcal{Z}(\mathrm{C}, \sigma)}{B \cdot 2\phi}$
    $\theta = \theta - \eta s \vec{z}$
end for
```

DP-ZO. We explain the steps of our algorithm while *emphasizing* the key differences from Definition 2.4 required to guarantee (ε, δ)-DP. We first sample a batch from the dataset with *Poisson sampling* (Balle et al., 2018) which allows us to use privacy amplification by subsampling. For each model parameter θ_i we want to update, we independently sample a perturbation z_i from a standard Gaussian distribution and scale it by a predetermined constant ϕ; we denote the full perturbation vector as $\phi\vec{z}$. Now we compute an approximation of the gradient by projecting it onto the random perturbation $\vec{z}$. That is, for a training sample x_i we compute the difference in scalar losses between $\theta + \phi\vec{z}, \theta - \phi\vec{z}$. Intuitively, this scalar tells us how much better one random step is than the other. We *clip* this scalar to limit the sensitivity . We *add noise* to the aggregation over samples in our training batch. Finally, we take a step in the direction of $\vec{z}$ by scaling our private step size by the expected batch size, perturbation constant ϕ, and the learning rate η.

Given a private scalar with bounded sensitivity, we can apply the classical Gaussian mechanism to release a privatized scalar with (ε, δ)-DP. The Gaussian mechanism is widely studied in privacy-preserving machine learning techniques like DP-SGD, in part because the best accounting techniques for the Gaussian mechanism (Dong et al., 2019; Gopi et al., 2021) are tight. However, the Gaussian mechanism can only provide (ε, δ)-DP. To avoid such a failure case in extremely sensitive applications, researchers often recommend using cryptographically small values of δ (Vadhan, 2017). Unfortunately, due to limitations of accounting methods, we currently cannot calculate the tight privacy of composition of sub-sampled Gaussian mechanism for values of δ smaller than 10^{-10}. Alternatively, we can resort to mechanisms that can obtain pure ε-DP. These mechanisms, such as Laplace mechanism, come with a guarantee that the mechanism will never fail catastrophically. However, due to large tails of the Laplace mechanism, it has never been a contender for high dimensional optimization.

Although it is possible to obtain pure DP with DP-SGD by adding Laplace noise scaled to the ℓ_1 sensitivity of the gradient, this is challenging for large models because the ℓ_1 sensitivity can be $\sqrt{d}$ times larger than the ℓ_2 sensitivity (and often is; see Appendix A.3), especially for billion-parameter LLMs. In contrast, DP-ZO only requires privatizing the loss. The one-dimensional private estimation of the step size is amenable to the Laplace mechanism, because the ℓ_p norms are equivalent. Specifically, *DP-ZO with the Laplace mechanism is the first method to achieve a reasonable privacy-utility trade-off under pure ϵ-DP for private fine-tuning of LLMs.* DP-ZO framework is flexible enough to be extended to other differential privacy mechanisms, broadening its applicability.

Privacy Analysis. As we consider multiple accounting methods with multiple previously proposed mechanisms, we give the overview of the analysis below and defer the full privacy analysis to Appendix C.

Theorem 3.1. *Algorithm 1 is (ε, δ)-DP.*

Proposition 3.2. *DP-ZO attains a convergence rate $\mathcal{O}(\sqrt{r}/\varepsilon n)$, where r is the effective rank of the problem.*

Malladi et al. (2023) proves the convergence rate of fine-tuning of language models with zeroth-order optimization is proportional to r instead of the model dimension. A concurrent work (Zhang et al., 2023) independently proposes DP-ZO and provides the convergence analysis for DP-ZO, that is independent of the model dimension in private training (Song et al., 2021; Li et al., 2022a). We discuss our work and Zhang et al. (2023) in Section 4.

Main results. We presents the main results of DP-ZO in Figures 2 to 4. We defer the full evaluations in Appendix A.

4. RELATED WORK

In this section we give an overview of the broader body of work privacy preserving large language models and private zeroth-order optimization method.

Privacy Preserving Large Language Models. Recent studies have leveraged DP-SGD to fine-tune large language models. Li et al. (2022b) provide methods for fine-tuning large language models with DP-SGD by ghost clipping to mitigate the memory burden of per-sample gradient clipping. Yu et al. (2022) report compelling results by only updating a sparse subset of the LLMs with parameter efficient fine-tuning (PEFT) methods such as LoRA (Hu et al., 2022). He et al. (2023) leverage group-wise clipping with adaptive clipping threshold and privately fine-tune the 175 billion-parameter GPT-3. Duan et al. (2023); Li et al. (2022b) also consider private prompt tuning by adding noise to the soft prompt (Li and Liang, 2021; Lester et al., 2021). Du et al. (2023) add non-i.i.d. noise from a matrix Gaussian distribution to directly perturb embedding in the forward pass of language models. With the emergence in-context learning of large language models (Brown et al., 2020), recent works (Duan et al., 2023; Wu et al., 2024; Tang et al., 2023b) study private in-context learning of large language models without fine-tuning.

Private Zeroth-order Optimization. Most recently, a concurrent work (Zhang et al., 2023) also considers the same DP-SPSA algorithm for zeroth-order optimization. Our method and Zhang et al. (2023) are functionally the same up to constants, and our work focuses on an empirical evaluation of the method, whereas Zhang et al. (2023) extends the convergence analysis of Malladi et al. (2023) to DP as shown in Appendix B of Zhang et al. (2023). There is a slight difference for the generation of random perturbation of Zhang et al. (2023) and our Algorithm 1. Zhang et al. (2023) uses the random unit vector for the perturbation and the convergence analysis is based on such set-up, whereas our perturbation is a normally distributed vector. Note that Algorithm 1 and 2 in Zhang et al. (2023) also scales the unit vector by the square root of the model dimension, so the two approaches are functionally the same. We reimplemented our perturbation method based on the algorithms in Zhang et al. (2023), and we obtain the comparable performance of with $(1, 10^{-5})$-DP for OPT-13B by LoRA finetuning on SQuAD as our main result in Table 1.

Zhang et al. (2024) study private zeroth-order nonsmooth nonconvex optimization. Their work incorporates two zeroth-order estimators to reduce variance and samples d (model dimension) i.i.d. estimators for each data point to achieve optimal dimension dependence. Zhang et al. (2024) leverage the tree mechanism (Dwork et al., 2010; Chan et al., 2011) on disjoint data to ensure the privacy cost of the algorithm. The main focus of our work is private fine-tuning of large language models and one estimator for each batch could successfully converge in this set-up. Therefore, we only need to privatize such scalar. We leave the investigation on the private zeroth-order for more than one estimators such as the variance reduction method proposed in Zhang et al. (2024) as future work.

Gratton et al. (2022) analyze the intrinsic privacy of the zeroth-order optimization for DP-ADMM (Huang et al., 2020) in distributed learning. Their work states that if the output of the zeroth-order method itself follows Gaussian distribution, the noise can be analyzed as the Gaussian mechanism and provide intrinsic privacy. However, this is merely stated as an assumption for lemma 1. To the best of our knowledge there is no work that proves that the zeroth-order gradient estimator can actually be analyzed as the sum of an unbiased gradient estimator and some Gaussian error term.

5. CONCLUSION

DP-SGD has been the de-facto private training method of the last decade. In this work we propose DP-ZO, a novel method for private fine-tuning that privatizes the zeroth-order update by adding noise to the difference in loss between two perturbations. DP-ZO's unique univariate privatization unlocks training larger models with better parallelism than DP-SGD. DP-ZO provides a strong privacy-utility trade-off across different tasks, model sizes, dataset sizes, and DP mechanisms. We anticipate that future work can further study these design choices, integrate more DP mechanisms into DP-ZO, and apply it to the vision domain.

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

B. Balle, G. Barthe, and M. Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Advances in Neural Information Processing Systems*, 2018.

R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473, 2014.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901, 2020.

Z. Bu, H. Wang, Z. Dai, and Q. Long. On the convergence and calibration of deep learning with differential privacy. *Transactions on Machine Learning Research*, 2023a. ISSN 2835-8856. URL `https://openreview.net/forum?id=K0CAGgjYS1`.

Z. Bu, Y.-X. Wang, S. Zha, and G. Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. In *Advances in Neural Information Processing Systems*, 2023b.

Z. Bu, Y.-X. Wang, S. Zha, and G. Karypis. Differentially private optimization on large model at small cost. In *Proceedings of the 40th International Conference on Machine Learning*, pages 3192–3218. PMLR, 2023c.

T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, (3):1–24, 2011.

S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

J. Dong, A. Roth, and W. J. Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019.

M. Du, X. Yue, S. S. M. Chow, T. Wang, C. Huang, and H. Sun. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, page 2665–2679, 2023.

D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, 2019.

H. Duan, A. Dziedzic, N. Papernot, and F. Boenisch. Flocks of stochastic parrots: Differentially private prompt learning for large language models. In *Advances in Neural Information Processing Systems*, 2023.

J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015. doi: 10.1109/TIT.2015.2409256.

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2014.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284, 2006.

C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, pages 715–724, 2010.

ffuuugor. fp16 support. GitHub issue, 2022. URL `https://github.com/pytorch/opacus/issues/377`.

A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient, 2004.

A. Ganesh, M. Haghifam, M. Nasr, S. Oh, T. Steinke, O. Thakkar, A. Guha Thakurta, and L. Wang. Why is public pretraining necessary for private model training? In *Proceedings of the 40th International Conference on Machine Learning*, pages 10611–10627. PMLR, 2023.

T. Gao, A. Fisch, and D. Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, 2021.

S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *arXiv preprint arXiv:1309.5549*, 2013.

S. Gopi, Y. T. Lee, and L. Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems*, pages 11631–11642, 2021.

C. Gratton, N. K. D. Venkategowda, R. Arablouei, and S. Werner. Privacy-preserved distributed learning with zeroth-order optimization. *IEEE Transactions on Information Forensics and Security*, 17:265–279, 2022. doi: 10.1109/TIFS.2021.3139267.

J. He, X. Li, D. Yu, H. Zhang, J. Kulkarni, Y. T. Lee, A. Backurs, N. Yu, and J. Bian. Exploring the limits of differentially private deep learning with group-wise clipping. In *The Eleventh International Conference on Learning Representations*, 2023.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong. Dp-admm: Admm-based distributed learning with differential privacy. *IEEE Transactions on Information Forensics and Security*, 15:1002–1012, 2020. doi: 10.1109/TIFS.2019.2931068.

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

X. Li, D. Liu, T. Hashimoto, H. A. Inan, J. Kulkarni, Y. Lee, and A. G. Thakurta. When does differentially private learning not suffer in high dimensions? In *Advances in Neural Information Processing Systems*, pages 28616–28630, 2022a.

X. Li, F. Tramèr, P. Liang, and T. Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022b.

X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

lxuechen. parameter sharing support. GitHub issue, 2022. URL `https://github.com/lxuechen/private-transformers/issues/12`.

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

I. Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security foundations Symposium (CSF)*, pages 263–275, 2017.

Y. Nesterov and V. G. Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

A. Panda, X. Tang, V. Sehwag, S. Mahloujifar, and P. Mittal. Dp-raft: A differentially private recipe for accelerated fine-tuning. *arXiv preprint arXiv:2212.04486*, 2022.

N. Ponomareva, H. Hazimeh, A. Kurakin, Z. Xu, C. Denison, H. B. McMahan, S. Vassilvitskii, S. Chien, and A. G. Thakurta. How to DP-fy ML: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, jul 2023. doi: 10.1613/jair.1.14649.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

O. Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *Proceedings of the 26th Annual Conference on Learning Theory*, pages 3–24. PMLR, 2013.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248, 2013.

S. Song, T. Steinke, O. Thakkar, and A. Thakurta. Evading the curse of dimensionality in unconstrained private glms. In *International Conference on Artificial Intelligence and Statistics*, pages 2638–2646. PMLR, 2021.

J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37:332–341, 1992.

X. Tang, A. Panda, V. Sehwag, and P. Mittal. Differentially private image classification by learning priors from random processes. In *Advances in Neural Information Processing Systems*, 2023a.

X. Tang, R. Shin, H. A. Inan, A. Manoel, F. Mireshghallah, Z. Lin, S. Gopi, J. Kulkarni, and R. Sim. Privacy-preserving in-context learning with differentially private few-shot generation. *arXiv preprint arXiv:2309.11765*, 2023b.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

F. Tramèr and D. Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

S. Vadhan. The complexity of differential privacy. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pages 347–450, 2017.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

J. T. Wang, S. Mahloujifar, T. Wu, R. Jia, and P. Mittal. A randomized approach for tight privacy accounting. In *Advances in Neural Information Processing Systems*, 2023.

T. Wu, A. Panda, J. T. Wang, and P. Mittal. Privacy-preserving in-context learning for large language models. In *International Conference on Learning Representations*, 2024.

A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021.

D. Yu, H. Zhang, W. Chen, and T.-Y. Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations*, 2021a.

D. Yu, H. Zhang, W. Chen, J. Yin, and T.-Y. Liu. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning*, pages 12208–12218. PMLR, 2021b.

D. Yu, S. Naik, A. Backurs, S. Gopi, H. A. Inan, G. Kamath, J. Kulkarni, Y. T. Lee, A. Manoel, L. Wutschitz, S. Yekhanin, and H. Zhang. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*, 2022.

L. Zhang, K. K. Thekumparampil, S. Oh, and N. He. Dpzero: Dimension-independent and differentially private zeroth-order optimization. *arXiv preprint arXiv:2310.09639*, 2023.

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

S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Y. Zhu, J. Dong, and Y.-X. Wang. Optimal accounting of differential privacy via characteristic function. In *International Conference on Artificial Intelligence and Statistics*, pages 4782–4817. PMLR, 2022.

A. EVALUATION

We first overview our experimental setup in Section A.1 and then evaluate the performance of DP-ZO in Section A.2. We find that DP-ZO provides a competitive privacy-utility tradeoff for conservative privacy budgets across multiple datasets, model architectures and model sizes under conservative privacy budgets. We also compare DP-ZO to DP-SGD in Section A.2 and show that DP-ZO achieves comparable performance to DP-SGD for the same model size. Furthermore, DP-ZO can seamlessly scale to large models such as OPT-66B, and the performance of DP-ZO increases with model size. In Section A.3 we first characterize DP-ZO under different few-shot settings. We then show that DP-ZO is robust to different mechanisms for noise addition. Specifically, DP-ZO with the Laplace mechanism is the first method to achieve a non-trivial privacy-utility trade-off under pure ε-DP.

A.1. Experimental Setup

When we report results, we report the metric of interest (F1 score or accuracy) and standard deviation averaged across 5 independent runs with different random seeds. We detail the full hyperparameter searches in Appendix F.

Datasets. We mainly consider three different benchmark NLP tasks: SQuAD (Rajpurkar et al., 2016) and DROP (Dua et al., 2019) for text generation, and SST2 (Socher et al., 2013) for text classification. Although all these datasets have very different dataset sizes, we consider the *few-shot* setting for all these datasets where we sample 1000 points for each dataset. Fine-tuning LLMs with $O(n = 1000)$ samples is a standard setting in the NLP community (Gao et al., 2021; Malladi et al., 2023) because we are generally interested in the few-shot abilities of LLMs (Brown et al., 2020). This represents a departure from prior works that privately finetune LLMs; Yu et al. (2022); Li et al. (2022b); Yu et al. (2021b) use the entire training dataset of SST2 that has about 65,000 examples. It is well known that the privacy-utility tradeoff improves greatly with more data (Tramèr and Boneh, 2021). It is straightforward to see that our setting with datasets of the size $n = 1000$ with $\delta = 10^{-5}$ is simultaneously more challenging and more aligned with real-world usecases than previous works in DP finetuning of LLMs. *Despite the increased difficulty of our few-shot setting as compared to prior work, our results validate that DP-ZO realizes a strong privacy-utility trade-off.* We also ablates the training sample size from the few-shot to the full training set on the QNLI (Wang et al., 2019) dataset.

Models. We use models including the OPT (Zhang et al., 2022), Mistral (Jiang et al., 2023), and roberta-base (Liu et al., 2019). We present our main results (Table 1) using a pretrained OPT-13B model that is finetuned with LoRA (Hu et al., 2022); that is, we update $< 1\%$ of the total parameters. We include a range of ablation studies, including varying the model size, model architectures and amount of parameters to be updated, after we present the main results.

Privacy Budgets. We consider various privacy levels with $\varepsilon = [0.5, 1, 4]$ and fix $\delta = 10^{-5}$ for (ε, δ)-DP. We include the zero-shot $\varepsilon = 0$ baseline that does not incur any privacy loss because we evaluate the pretrained model directly without finetuning on private data. We also include the non-private $\varepsilon = \infty$ baseline that is trained without any DP guarantee. That is, we iterate over the shuffled dataset instead of doing Poisson sampling (replacing line 4), do not clip the step size (skipping line 11) and set $\sigma = 0$ (in line 13). We make these modifications because Poisson sampling and clipping are known to degrade performance, and we want to compare to the strongest possible nonprivate baseline.

A.2. Main Results

DP-ZO Provides a Strong Privacy-utility Tradeoff for Conservative Privacy Budgets. As shown in Table 1, across all three tasks and all εs, DP-ZO significantly improves upon the $\varepsilon = 0$ baseline, and only slightly degrades the performance compared to the non-private baseline. For SQuAD, even at $\varepsilon = 0.5$, DP-ZO can still achieve 80.10%, that significantly outperforms $\varepsilon = 0$ baseline (46.23%). The gap between $\varepsilon = 0.5$ and $\varepsilon = \infty$ is about 6.75%. By increasing ε from 0.5 to 4, this gap can be further reduced to 3%. For DROP and SST2, DP-ZO (Gaussian) achieves comparable performance as the non-private baseline at $\varepsilon = 4$.

DP-ZO Scales to Large Models. In Table 2 we show that DP-ZO continues improving as the model size increases from 1.3B to 66B. Due to space constraints, we provide the non-private ($\epsilon = \infty$) performance of all models and methods in Appendix G. Table 2 shows an promising insight: *as the model size and nonprivate performance increase, the gap in performance between private and nonprivate models shrinks.* Specifically, the gap for OPT-1.3B is 5.68% (80.97% at $\varepsilon = \infty$ reduced to 75.29% under $\varepsilon = 1$). But this gap shrinks to just 2.11% for OPT-66B, where the private performance at $\epsilon = 1$ is 85.38% compared to 87.49% non-privately. Our findings suggest that DP-ZO scales to large models not only because it is compatible with existing pipeline without much additional engineering effort but also because the utility drop due to privacy is smaller as the model size increases.

Comparison with DP-SGD. We compare DP-ZO to differentially private stochastic gradient descent (DP-SGD) (Abadi et al., 2016) which has been applied to fine-tune LLMs with full parameter fine-tuning (Li et al., 2022b) and with LoRA (Yu et al., 2022; He et al., 2023). Recall that DP-ZO is compatible out-of-the-box with mixed precision training and GPU parallelism, enabling us to fine-tune OPT-

Table 1. Main results with 1000 training samples for each dataset. OPT-13B model with LoRA fine-tuning. DP-ZO (G) is DP-ZO instantiated with the Gaussian mechanism. $\delta = 10^{-5}$. The $\varepsilon = \infty$ by ZO is 86.85 for SQuAD, 33.22 for DROP, and 93.69 for SST2. The $\varepsilon = 0$ baseline, i.e., directly doing model evaluation without training, is 46.23 for SQuAD, 14.64 for DROP, and 58.83 for SST2.

Task	SQuAD			DROP			SST2		
Task type	generation (metric: F1)						classification (metric: accuracy)		
Method	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 4$	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 4$	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 4$
DP-ZO(G)	$80.10_{0.63}$	$82.28_{0.84}$	$83.87_{0.50}$	$28.39_{0.82}$	$30.30_{0.51}$	$31.99_{0.51}$	$85.41_{2.91}$	$91.19_{0.90}$	$92.59_{0.30}$

Table 2. DP-ZO (Gaussian) and DP-SGD with full parameter and LoRA fine-tuning on SQuAD with 1000 training samples across different model sizes. $(1, 10^{-5})$-DP. '-' means the approach did not scale with straightforward implementation; Appendix B details the additional engineering required to scale DP-SGD to larger models. '- -' for DP-ZO means the results are omitted due to limited computational resources. Due to limited computing resources, this table does not include the standard deviation for OPT-66B model.

Method	OPT-1.3B	OPT-2.7B	OPT-6.7B	OPT-13B	OPT-30B	OPT-66B
DP-ZO-LoRA (Gaussian)	$75.29_{0.90}$	$80.34_{1.14}$	$81.34_{1.04}$	$82.28_{0.84}$	$82.48_{0.83}$	$84.12_{1.01}$
DP-SGD-LoRA	$75.39_{0.33}$	$79.42_{0.57}$	$79.53_{0.52}$	$82.14_{0.18}$	-	-
DP-ZO-Full (Gaussian)	$72.84_{1.03}$	$77.25_{0.27}$	$79.06_{0.67}$	$82.16_{0.41}$	- -	- -
DP-SGD-Full	$75.50_{0.89}$	$79.81_{0.64}$	-	-	-	-

66B. As we show in Appendix B, it is significantly more challenging to integrate DP-SGD with these techniques, and furthermore, DP-SGD requires more memory than DP-ZO to store activations and compute per-sample gradients. As a direct result, DP-SGD cannot directly scale past 2.7B with full fine-tuning or 13B with LoRA without additional implementation effort for multi-GPU training, while DP-ZO can scale seamlessly to larger models. In Table 2 we present comparisons between DP-ZO and DP-SGD with full parameter finetuning and LoRA. With the same model size, DP-ZO achevies comparable performance as DP-SGD as by LoRA finetuning, i.e., both DP-ZO and DP-SGD achieves 82% on OPT-13B models. The best performance by DP-ZO is 85.38% by OPT-66B finetuned with LoRA. This is $\approx 3\%$ better than the best performance of DP-SGD in Table 2 that is 82.14% by OPT-13B with LoRA.

DP-ZO Provides a Strong Privacy-utility Tradeoff across Different Model Architectures. Table 1 and Table 2 show that DP-ZO achieves the comparable performance as DP-SGD on OPT models. We also run experiments on SQuAD with Mistral-7B model. DP-ZO and DP-SGD achieves comparable performance at $\varepsilon = 1$.

DP-ZO with Pure ε-DP. To the best of our knowledge, DP-ZO (Laplace) is the first method that achieves a non-trivial privacy-utility tradeoff under pure ε-DP under a conservative privacy budget like $\varepsilon = 4$ on large language models.

Table 3. Pure ε-DP by DP-ZO (Laplace), SQuAD with 1000 training samples. OPT-13B with LoRA fine-tuning. The $\varepsilon = \infty$ performance with ZO is 86.85% and the $\varepsilon = 0$ baseline is 46.23%.

ε	$\varepsilon = 4$	$\varepsilon = 10$	$\varepsilon = 15$
DP-ZO (Laplace)	$73.52_{1.04}$	$76.75_{1.39}$	$78.82_{1.57}$

Table 4. Pure ε-DP by DP-ZO (Laplace) and DP-SGD (Laplace) at $\varepsilon = 4$, SQuAD with 1000 training samples. OPT-13B with LoRA fine-tuning.

Method	DP-ZO	DP-SGD
F1	$73.52_{1.03}$	$47.25_{0.79}$

In Table 3, DP-ZO (Laplace) can significantly improve upon $\varepsilon = 0$. Given a budget $\varepsilon = 4$, which some prior work has considered reasonable (Ponomareva et al., 2023), DP-ZO (Laplace) can obtain 73.52% on SQuAD. When increasing $\varepsilon = 4$ to $\varepsilon = 15$, DP-ZO (Laplace) can obtain 78.82% on SQuAD. Note that the l_1 sensitivity required for Laplace mechanism makes it hard to DP-SGD to achieve comparable performance as DP-ZO because the gradients in DP-SGD have high dimension. In contrast, DP-ZO only requires privatizing a scalar value. Table 4 shows that DP-SGD with l_1 norm clipping and Laplace noise does not converge and only achieves 47.25%, that is only marginal improvement upon the zero-shot performance. Besides, the few-shot setting poses a unique challenge for obtaining strong performance under conservative privacy budgets. Acquiring more data enhances privacy amplification and reduces the amount of noise we need to add to achieve a target ε-DP guarantee.In particular, in Table 5 we find that increasing the number of training examples from 1000 to 5000 improves performance at $\varepsilon = 4$ from 73.52% to 79.89%, although the improvement of non-private performance at $\epsilon = \infty$ by increasing training samples from 1000 to 5000 is insignificant. *DP-ZO (Laplace) presents a new direction for obtaining pure ε-DP guarantees.*

Empirical privacy evaluation. We conducted membership inference attacks Shokri et al. (2017) on DP-ZO on SQuAD using OPT-13B model. We show that such MIAs can only achieve attack AUC around 0.50, that is closed to random

Table 5. Pure ε-DP by DP-ZO (Laplace) at $\varepsilon = 4$, SQuAD with different training samples. OPT-13B with LoRA fine-tuning. The $\varepsilon = \infty$ by ZO is 86.85% and 86.92% for 1000 and 5000 samples respectively. The $\varepsilon = 0$ baseline is 46.2%.

n-shot	$n = 1000$	$n = 5000$
DP-ZO (Laplace)	$73.52_{1.04}$	$79.89_{0.49}$

guess. This empirical privacy evaluation shows that DP-ZO can effectively protect models from privacy attacks like MIAs.

A.3. Analysis

In this section, we vary the amount of training data that we sample and the choice of DP mechanism in DP-ZO.

Characterizing Differentially Private Few-Shot Learning. Although it is known that private learning requires more data than non-private learning (Bassily et al., 2014), prior work has not characterized this improvement for fine-tuning language models. In Figure 5 and Table 6 we vary the number of training samples n around the $n = 1000$ setting in the main results while keeping $\delta = 10^{-5}$ fixed for all choices of n. Table 6 shows that DP-ZO can achieve non-trivial performance in few-shot settings under conservative privacy guarantees. Furthermore, we find that while increasing the amount of training data by 10× barely increases non-private performance, it increases private performance by $\approx 6\%$ ($n = 500$ vs. $n = 5000$). While non-private few-shot learning can succeed by just memorizing the training data, Figure 5 indicates that the convergence rate for different shots for private few-shot learning is different. With the proliferation of pretrained models, we anticipate that *privately fine-tuning downstream tasks in the few-shot setting will be more aligned with real-world use cases.*

Table 6. Ablation of DP-ZO (Gaussian) for different n training samples on SQuAD dataset. $(1, 10^{-5})$-DP. OPT-13B with LoRA finetuning.

n-shot	$n = 250$	$n = 500$	$n = 1000$	$n = 5000$
$\varepsilon = 1$	$74.86_{0.74}$	$78.25_{2.38}$	$82.28_{0.84}$	$84.29_{0.92}$
$\varepsilon = \infty$	86.40	86.53	86.85	86.92

Characterizing the effect of data size. So far we have studied DP-ZO under the few-shot setting, that is a challenging than accessing the full dataset size from the privacy-perspective. We now analyze the effect of data size on DP-ZO and DP-SGD by varying data size from 250 shot to the full show size. Specifically, we conduct experiments on QNLI dataset, that has about 100,000 data point in the full training set, by training Roberta-base models. We observe the trend when DP-ZO incurs a utility drop compares to DP-SGD within the increase of dataset size. Our analysis for this is that the convergence slowness in zeroth-order method is more significantly within more data due to the inefficiency use of data. Potential studies on improving the convergence rate in zeroth-order method could potentially benefit DP-ZO by improving the efficiency use of data.

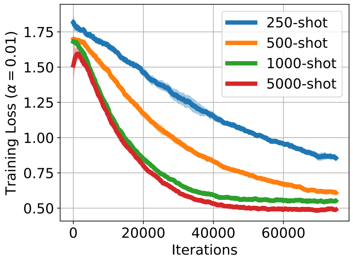

Figure 5. (Smoothed) training loss. $n = 5000$ has better convergence rate compared to $n = 250$.

Different Noise Mechanisms for (ε, δ)-DP. We now relax the privacy guarantee provided by the Laplace mechanism to approximate (ε, δ)-DP. In Table 7, we compare DP-ZO instantiated with the Laplace and Gaussian mechanisms. DP-ZO (Gaussian) outperforms DP-ZO (Laplace) for strict privacy budgets such as $\varepsilon = 0.5$ because it enjoys tighter accounting (Gopi et al., 2021) and lower variance (Dwork and Roth, 2014). These advantages are less significant for larger privacy budgets; for $\varepsilon = 4$, the gap between DP-ZO (Gaussian) and DP-ZO (Laplace) is within 1%.

Table 7. DP-ZO with different DP mechanism. SQuAD with 1000 training samples. $\delta = 10^{-5}$. G is for Gaussian and L is for Laplace.

ε	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 4$
DP-ZO (G)	$80.10_{0.63}$	$82.28_{0.84}$	$83.87_{0.50}$
DP-ZO (L)	$77.58_{0.81}$	$80.49_{0.63}$	$82.94_{0.69}$

Our ablation study on the DP-ZO with laplace for ε-DP and the comparisons of Laplace and Gaussian mechanisms for (ε, δ)-DP shows that DP-ZO provides a strong privacy-utility trade-off under different DP mechanisms while DP-SGD suffers from Laplace mechanisms for (ε, δ)-DP, which opens the new opportunity for the synergy between DP mechanisms and large language models.

Memory Analysis. We compare the GPU cost of DP-ZO, to DP-SGD, for both full parameters trainable, as well as parameter-efficient fine-tuning (PEFT) methods including DP-LoRA, DP-BiTFiT on SQuAD using OPT-13B models. We show our DP-ZO costs less GPU memory consumption as a result that DP-ZO will not generate the vectors for gradients.

B. DISCUSSION

In Appendix A.2 we showed that DP-ZO obtains competitive privacy-utility tradeoff. Now we examine the amount of engineering effort necessary to scale DP-SGD to larger models, a topic on which many papers have been written (Bu et al., 2023a;b; Yousefpour et al., 2021; Li et al., 2022b; He et al., 2023). We find that DP-ZO *seamlessly scales to larger models* and believe its simplicity presents a compelling alternative to DP-SGD for practitioners.

DP-SGD. Differentially Private Stochastic Gradient Descent (DP-SGD) (Song et al., 2013; Abadi et al., 2016) is the standard privacy-preserving algorithm to train models on private data, with an update rule given by $w^{(t+1)} = w^{(t)} - \frac{\eta_t}{|B|}\left(\sum_{i \in B} \frac{1}{\mathrm{c}} \mathtt{clip}_{\mathrm{c}}(\nabla \ell(x_i, w^{(t)})) + \sigma\xi\right)$ where the changes to SGD are the per-sample gradient clipping $\mathtt{clip}_{\mathrm{c}}(\nabla \ell(x_i, w^{(t)})) = \frac{c \times \nabla \ell(x_i, w^{(t)})}{\max(c, ||\nabla \ell(x_i, w^{(t)})||_2)}$ and addition of noise sampled from a d-dimensional Gaussian distribution $\xi \sim \mathcal{N}(0, 1)$ with standard deviation σ. DP-SGD is the marquee algorithm for privacy-preserving machine learning, but it requires implementing per-example gradient clipping. This creates a slew of challenges for deploying DP-SGD.

Computational and Memory Challenges in DP-SGD. DP-SGD requires the computation of per-example gradients, which can be naively implemented by storing each gradient in the batch separately. This approach inflates the memory overhead by a factor of B, where B is the batch size. Tensorflow Privacy avoids this issue by clipping microbatches rather minibatches, which does not slow down training but increases the noise added and therefore hurts utility. Jax can automatically vectorize the per-sample gradient computation, but training is still slowed down. Recently, specialized libraries have been developed that instead analytically compute the norm of the gradients for different layers (Li et al., 2022b; Bu et al., 2023c). This requires actually implementing the computation, which is challenging for new layers. If a practitioner wants to train LLaMA2 and efficiently compute the per-example gradient norms, they would first have to derive the analytical formula for the norm of Grouped-QueryAttention (Touvron et al., 2023), which can represent a nontrivial amount of engineering. This complexity is further compounded when considering models with parameter sharing, such as the OPT networks we use in our experiments, as this is generally not compatible with analytical norm clipping methods (lxuechen, 2022). Although network architectures exist that do not require parameter sharing, the best pretrained models (Zhang et al., 2022; Touvron et al., 2023) use parameter sharing because it makes the model use the same representation for predicting tokens at the next step as well as for decoding.

B.1. DP-ZO Scales Seamlessly

In order to train models as large as OPT-66B, whose parameters cannot be loaded into memory on a single A100 GPU, we need to implement some form of parallelism across GPUs. We now discuss how easy this is for DP-ZO (in the simplest form, just running DP-ZO on a machine with 2 GPUs will prompt HuggingFace to implement naive model parallelism) and how challenging it can be for approaches that require per-example gradient clipping such as DP-SGD.

Data Parallelism in DP-ZO. To synchronize model state between GPUs in data-parallel-DP-ZO, we just transfer the random seed and its corresponding fp16 scalar step size; this is just a few bytes. However, first-order approaches such as DP-SGD require the transfer of gradients across devices to update all the models, necessitating expensive allgather and reduce operations. This communication overhead is $1.5d$ in PyTorch FSDP, where d is the size of the model.

DP-ZO Does Not Store Gradients. DP-ZO does not store activations or gradients in the forward pass, thus conserving memory. DP-SGD needs to store the activations at each GPU under pipeline parallelism to clip the per-example gradient (He et al., 2023), which will fill up the GPU memory and limit new microbatches from being processed.

DP-ZO Easily Integrates With Mixed Precision Training. All DP-ZO experiments use mixed precision; we load the model and compute the loss in half precision. Mixed precision training is much faster than full precision and also has a smaller memory footprint; these factors contribute to DP-ZO's ability to train much larger models. By contrast, implementing DP-SGD with half precision is challenging. Yu et al. (2022) do not modify their implementation of DP-SGD to support half precision and report worse performance as a result. Li et al. (2022b) detail an algorithm that interleaves loss scaling with gradient clipping to avoid underflow while maintaining utility. This implementation is nontrivial and as of today is not available in Opacus (ffuuugor, 2022).

DP-ZO is Storage and Communication-Efficient Even After Training Has Completed. DP-ZO offers significant advantages in terms of storage and communication efficiency, especially beneficial for bandwidth-constrained environments like edge devices. Unlike traditional methods where the difference in model parameters $\theta_0 - \theta_f$ is shared—which could amount to multiple gigabytes for large models—DP-ZO allows for the storage and transmission of a sequence of updates. This sequence is represented as an array of tuples $[(\textsc{seed}_0, 0.54), \cdots, (\textsc{seed}_f, -0.14)]$, where each tuple contains a seed and a step size, taking up only 4 bytes. Even for 1×10^4 fine-tuning iterations, this array would require less than 1MB of storage, representing a substantial reduction in both storage and communication overhead. We can apply these weight differences to a model by simply iterating over the array, sampling from the PRNG using the given seed, scaling that random vector, and applying it to the current model parameters. This procedure is highly efficient, as it involves only sequential memory accesses and scalar floating-point operations.

C. Privacy Analysis

Proposition C.1 (Basic Composition theorem (Dwork and Roth, 2014))**.** *If M_1 is $(\varepsilon_1, \delta_1)$-DP and M_2 is $(\varepsilon_2, \delta_2)$, then the adaptive composition of M_1 and M_2 is $(\varepsilon_1+\varepsilon_2, \delta_1+\delta_2)$-DP.*

Proposition C.2 (Privacy Amplification via Subsampling (Balle et al., 2018))**.** *If M is (ε, δ)-DP, then the subsampled mechanism with sampling rate p obeys (ε', δ')-DP with privacy parameters $\varepsilon' = \log(1 + p(e^{\varepsilon} - 1))$ and $\delta' = p\delta$.*

DP-ZO can be instantiated with different noise mechanisms. In this subsection we provide privacy analysis for the Gaussian mechanism and Laplace mechanism.

Gaussian Mechanism As outlined in Line 11, the ℓ_2 sensitivity of Algorithm 1 is C and we are adding $\mathcal{N}(0, \mathrm{C}^2\sigma^2)$ noise to the estimated loss. We analyze the composition of subsampled Gaussians with the privacy loss variable accounting approach of Gopi et al. (2021).

Laplace Mechanism Laplace mechanism can gives a pure DP guarantee of $\delta = 0$ which can be of interest in some scenarios. Here we first analyze the pure ε-DP guarantee provided by laplace mechanism and then provide the analysis for approximate (ε, δ)-DP analysis.

Pure ε-DP by Laplace mechanism. We use data in a single batch instead of all training data to compute the gradients in each updates. For the privacy analysis for Laplace mechanism in Algorithm 1, when we sample each batch in the poisson manner, we could leverage Proposition C.2 to compute the private amplification by subsampling. We first analyze the privacy cost for one step by the Laplace mechanism. At each step, we sample a new batch of data with the sample rate of $p = B/|\mathcal{D}|$. As outlined in Line 11, the ℓ_1 sensitivity of Algorithm 1 is C. By Section 2.1, the privacy cost at one step would cost $(1/\sigma, 0)$-DP on this batch. By Proposition C.2, the privacy cost at one step would cost $(\log(1 + p \cdot (e^{1/\sigma} - 1)), 0)$-DP on the full dataset $\mathcal{D}$. By Proposition C.1, the privacy cost of Algorithm 1 instantiated with Laplace mechanism satisfies $(T \cdot \log(1 + p \cdot (e^{1/\sigma} - 1)), 0)$-DP.

We provide the privacy parameters we used for pure ε-DP by the Laplace mechanism in Table 8.

Approximate ε-DP by Laplace mechanism. We can also get tighter composition of ε with relaxation to $\delta > 0$. The most straight forward way is to instantiate the PRV of random response with $(\log(1 + p \cdot (e^{1/\sigma} - 1)), 0)$ because the dominating pair for random response is a dominating pair for the pure DP mechanism. Then, we can use the numerical composition of Pure DP PRV by Gopi et al. (2021). Note that this method is agnostic to the DP mechanisms used for pure ε-DP. We now provide a more fine-grained privacy analysis for the laplace mechanism. Specifically, we could compute the privacy cost of composition for the Laplace mechanism by Monte Carlo based DP accountant (Wang et al., 2023). Note that since we are dealing with scalar values, our mechanism in each iteration will be a one dimensional Laplace mechanism. Let b be the scale of Laplace noise, p the sub-sampling rate, and assume the sensitivity is 1, and assume we are doing composition for T iterations, each iteration with sampling rate p. By Zhu et al. (2022) we know that the pair of distribution (P, Q) dominating pair for a single dimensional Laplace mechanism, where P and Q are distributed according to the following pdfs,

$$f_P = \frac{1}{2b}\exp(-|x|/b) \quad \text{and} \quad f_Q = \frac{1}{2b}\exp(-|x-1|/b).$$

Therefore, $(P, (1-p) \cdot P + p \cdot Q)$ is the dominating pair for the sub-sampled Laplace. We plug this into the standard Monte-Carlo accountant of Wang et al. (2023) (without importance sampling, see Algorithm 2 and Theorem 10 in Wang et al. (2023)) while using 10^{10} samples to calculate the δ at a given value of ϵ. Also, using the analytical accountant explained above, we always make sure that $\mathbb{E}[\hat{\delta}^2_{MC}]$ is bounded by 10^{-8} (We use the fact that $\mathbb{E}[\hat{\delta}^2_{MC}]$ is bounded by $\mathbb{E}[PRV^2]$ and the fact the PRV is always bounded for Laplace mechanism.). This ensures that the error of our estimation of δ is at most 10^{-8} with probability at least $1 - 10^{-5}$. Putting all together, for all reported values of ϵ, our δ is bounded by 10^{-5}, with probability at least 0.99999. This privacy analysis is tighter with ε is high compared to the former analysis which uses the pure DP PRV accountant. This is consistent with the intuition. As we increase the distance between the Laplace dominating pairs, the probability of sampling points from the area between the centers increases. And that is where the Laplace Mechanism is different from the Randomized Response. We present the accounting results for the Laplace method to achieve (ε, δ)-DP by these two accounting methods in Table 9. Table 9 shows that the Monte Carlo based DP accountant can give tighter analysis for the Laplace mechanism for (ε, δ)-DP than the pure ε-DP PRV method.

Table 8. ε-DP by Laplace. BSZ=20, Steps=2000.

σ	$\|D\|$	ε
10.5	1000	4
4.5	1000	10
3.2	1000	15
2.5	5000	4

Table 9. (ε, δ)-DP guarantee for Laplace. $\delta = 10^{-5}$. $|\mathcal{D}| = 1000$. BSZ=16, Steps=75000.

σ	ε (by Monte-Carlo)	ε (by pure-DP PRV)
30.8	0.5	0.51
16.3	1	1.04
4.6	4	4.70

D. Implementation Details

We follow Malladi et al. (2023) and provide the memory-efficient version of DP-ZO in Algorithm 2. Algorithm 2 enjoys the benefit that it does not incur additional GPU memory cost compared to inference.

Algorithm 2 Differentially Private-ZO (GPU memory efficient version. Adapted from Malladi et al. (2023))

```
Model parameters $\theta$, dataset $\mathcal{D}$, learning rate $\alpha$, perturbation scale $\phi$, random seed $s$, weight decay $\lambda$, noise scale $\sigma$, noising mechanism $\mathcal{Z}$, clipping threshold $C$, expected batch size $B$ and sampling rate $p = B/|\mathcal{D}|$. Lines with * are DP modifications.
procedure DP-ZO$((\theta, \mathcal{D}, \epsilon, \sigma, T, s, \phi, C, \alpha))$
    for $t \in 1, \ldots T$ do
        Poisson samples $\mathcal{B}$ from dataset D with sampling rate $p$ *
        $\theta \leftarrow$ PerturbParameters$(\theta, \phi, s)$
        Compute per-sample loss $\mathcal{L}_1(\theta, \mathcal{B})$*
        $\theta \leftarrow$ PerturbParameters$(\theta, -2\phi, s)$
        Compute per-sample loss $\mathcal{L}_2(\theta, \mathcal{B})$*
        $\theta \leftarrow$ PerturbParameters$(\theta, \phi, s)$
        Compute difference in loss $\mathcal{L} = \mathcal{L}_1 - \mathcal{L}_2$
        Clamp $\mathcal{L}$ between $-C$ and $C$*
        $g = \frac{\sum_{i \in \mathcal{B}} L + \mathcal{Z}(C, \sigma)}{B * 2\phi}$ *
        Reset random number generator with seed $s$
        for $\theta_i \in \theta$ do
            $z \sim \mathcal{N}(0, 1)$
            $\theta_i \leftarrow \theta_i - \alpha * g * z$
        end for
    end for
end procedure
procedure PERTURBPARAMETERS$((\theta, \phi, s))$
    Reset random number generator with seed $s$
    for $\theta_i \in \theta$ do
        $z \sim \mathcal{N}(0, 1)$
        $\theta_i \leftarrow \theta_i + \phi z$
    end for
end procedure
```

E. Design Choices

Algorithm 1 outlines our DP-ZO that estimates the gradients via privatized loss value without backpropagation. In this subsection, we provide several design choices for Algorithm 1.

Definition 2 (n-SPSA Gradient Estimator) The n-SPSA gradient estimate averages $\nabla L_b(\theta; B)$ over n randomly sampled z. We can write this in vector notation, dropping the normalizing constants for succinctness.

$$g_i = L(\theta + \epsilon z_i; B) - L(\theta - \epsilon z_i; B) (\text{projected gradient for each } i)$$

$$\mathbf{Z} = [z_1, z_2, ..., z_n] (\text{matrix whose columns are the } z \text{ vectors})$$

$$\mathbf{g} = [g_1, g_2, ..., g_n] (\text{vector of projected gradients})$$

Then the n-SPSA gradient estimate can be written as:

$$\nabla L_n(\theta; B) = \mathbf{g} \cdot Z \quad (2)$$

How Many Gradients to be Estimated in a Model Update. Algorithm 1 estimates the gradients once. As outlined above, SPSA can be extended to n-SPSA gradient estimator and n-SPSA can improve the performance in the non-private setting (Malladi et al., 2023). Here we discuss our design choice of why we choose $n = 1$ in Algorithm 1.

- Estimate the average. Previous work (Malladi et al., 2023) shows that averaged estimation helps the non-private setting. In a private setting, we have to privatize the gradient estimation. Here we discuss our initial design of the privatized n-SPSA gradient estimation. For the sampled batch, assuming we are adding the Gaussian noise $\mathcal{N}(0, \mathrm{C}^2\sigma^2)$ for 1-SPSA. Then for n-SPSA, to ensure we have the same privacy cost as 1-SPSA, we need to add $\mathcal{N}(0, n \cdot \mathrm{C}^2\sigma^2)$ to each gradient estimation and finally average the n gradients. Our privacy analysis follows the n-fold composition of Gaussian mechanism (Corollary 3.3 in Gaussian differential privacy (Dong et al., 2019)). Our initial experiment result shows that our current analysis for n-SPSA noise addition does not make n-SPSA improve in the private setting compared to 1-SPSA. We leave the improvement in tighter analysis for private n-SPSA as future work and use 1-SPSA to conduct experiments.

The Type of Noise for DP. As discussed in Section 3, Algorithm 1 can be incorporated in different noise mechanisms. We focus on the Gaussian noise mechanism and the Laplace mechanism in this work. The Gaussian noise mechanism has been widely studied in previous literature both for privacy analysis and empirical performance in DP-SGD (Abadi et al., 2016; Mironov, 2017; Dong et al., 2019). The Laplace mechanism, though less studied for privacy-preserving machine learning, can provide pure DP while the Gaussian mechanism can only provide approximate DP. We have provided the privacy analysis in Section C.

F. Hyperparameter Search

Our experiments are based on the open-source code[1] of Malladi et al. (2023). We provide the prompts we use in Table Table 10. In this section, we first provide several initial results for hyperparameter search on clipping threshold and finally present the hyperparameter tables. We also provide an initial study to systematically evaluate the interplay between batch size and training iterations for DP-ZO.

Table 10. The prompts of the datasets we used for DP-ZO.

Dataset	Type	Prompt
SQuAD	QA	Title: `<title>` Context: `<context>` Question: `<question>` Answer:
DROP	QA	Passage: `<context>` Question: `<question>` Answer:
SST-2	classification	`<text>` It was terrible/great

Different Clipping Threshold. Li et al. (2022b); De et al. (2022) recommend small clipping C threshold for DP-SGD training. For example, Li et al. (2022b) use $C = 0.1$ for training language models. We therefore study the effect of different clipping threshold and present the results in Table 11. We find that while $C = 1$ performs significantly worse, setting C as 0.1, 0.05, 0.01 are within the 2% performance gap. We therefore choose $C = 0.05$.

Table 11. Different clipping C. $\sigma = 15.9$. batch size=16, 10,000 steps. $\varepsilon = 0.35$.

	Clip=1	Clip=0.1	Clip=0.05	Clip=0.01
F1	66.04	74.26	76.81	75.39

Hyperparameter for DP-ZO (Gaussian) in Main Results. We present the hyperparameter for DP-ZO (Gaussian) in Table 12 and Table 13.

Table 12. Hyperparameter search for DP-ZO in main results Table 1.

$\vert\mathcal{D}\vert$	1000
Steps T	75000
Clipping C	0.05
Batch size	16
σ	30.9 for $\varepsilon = 0.5$, 16.4 for $\varepsilon = 1$, 4.8 for $\varepsilon = 4$
learning rate	[5e-6, 1e-5, 2e-5, 5e-5, 1e-4]
LoRA rank	8

Table 13. Hyperparameter search for DP-ZO with full parameter fine-tuning in Table 2.

$\vert\mathcal{D}\vert$	1000
Steps T	10000
Clipping C	0.05
Batch size	16
σ	11.47 for $\varepsilon = 0.5$, 6.08 for $\varepsilon = 1$, 1.88 for $\varepsilon = 4$
learning rate	[2e-7, 5e-7, 1e-6, 2e-6, 5e-6]

Hyperparameter for DP-SGD. We present the hyperparameter search for DP-SGD in Table 14.

Table 14. Hyperparameter search for DP-SGD in Table 2.

$\vert\mathcal{D}\vert$	1000
Steps T	200
Clipping C	0.1
Batch size	64
σ	6.60 for $\varepsilon = 0.5$, 3.59 for $\varepsilon = 1$, 1.28 for $\varepsilon = 4$
learning rate	[1e-4, 2e-4, 5e-4, 1e-3, 2e-3] for LoRA fine-tuning. [1e-5, 2e-5, 5e-5, 1e-4, 2e-4, 5e-4] for Full fine-tuning.
LoRA rank	8

Hyperparamter for DP-ZO (Laplace). The hyperparameter search for DP-ZO (Laplace) is similar to DP-ZO (Gaussian).

Ablation on Batch Size and Steps. In Table 15 and Table 16, we did an initial study to systematically evaluate the interplay between batch size and training iterations by varying batch size in [16,32,64,128] and steps in [10000, 2000, 40000, 80000]. Similar to main results, we run 5 independent runs for each setting and compute the average of 5 runs. This ablation is by OPT-13B on SQuAD dataset with LoRA fine-tuning. Table 15 and Table 16 show that increasing steps T improves the performance more than increasing the batch size. We also did ablation study on T in [200, 400, 800, 1600] for DP-SGD (and did not observe significant improvements in DP-SGD) to ensure the

[1] https://github.com/princeton-nlp/MeZO.

fair comparison of DP-SGD and DP-ZO. Taking the computation limitation into consideration, we set $T = 75000$ and BSZ=16 for main results in Table 1. We leave more investigation on the batch size and steps for DP-ZO, such as variance reduction method, as future work.

Table 15. $T = 10000$, Varying batch size.

	BSZ=16	BSZ=32	BSZ=64	BSZ=128
F1	81.35	81.63	81.47	81.72

Table 16. Batch size=16. Varying steps T.

T	10000	20000	40000	80000
F1	81.35	81.65	81.42	82.52

Computation Cost. DP-ZO for OPT-13B models on SQuAD datasets takes around 4hrs for 10000 steps. DP-SGD for OPT-13B models on SQuAD datasets takes around 4hrs for 200 steps. When increasing T or B in DP-ZO, the training time scales proportionally to the scaling factor. Future work includes how to reduce the computation time of DP-ZO, e.g., by variance reduction method to improve the convergence rate.

G. Ablation on Model Size

Section A.2 shows that DP-ZO scales to larger models and provides the results of DP-ZO for model size varying from 1.3B to 66B parameters in Table 2. Here we provide the full results of DP-ZO finetuned with LoRA at $\varepsilon = 1$, with model size ranging from 1.3B to 66B. We also include the $\varepsilon = 0$ and $\varepsilon = \infty$ baseline as a reference in Table 17.

Table 17 shows the full trend of DP-ZO with model size scaling from 1.3B to 66B, that is DP-ZO scales to larger models.

For OPT-1.3B, the gap between private and non-private baseline is 5.67. For OPT-66B, the non-private baseline is 87.49 and the gap between the private and non-private results is 2.11.

Table 17. Ablation of DP-ZO across different model sizes. $(1, 10^{-5})$-DP.

Model	OPT-1.3B	OPT-2.7B	OPT-6.7B	OPT-13B	OPT-30B	OPT-66B
$\varepsilon = 0$	27.20	29.89	36.48	46.23	46.53	48.13
$\varepsilon = 1$	$75.29_{0.90}$	$80.34_{1.14}$	$81.34_{1.04}$	$82.28_{0.84}$	$82.48_{0.83}$	85.38_{*}
$\varepsilon = \infty$	80.97	84.14	86.44	86.85	86.98	87.49