DIFFICULTY-AWARE REWEIGHTING FOR ZEROTH-ORDER FINE-TUNING OF LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning large language models (LLMs) is essential for downstream tasks, but backpropagation incurs prohibitive activation memory. Zeroth-order (ZO) optimization uses only forward passes, which provides a memory-efficient route to LLM adaptation on resource-constrained devices. Despite substantial progress on ZO optimizers and query schemes, the data-dependent characteristics of ZO gradient estimation remain under-explored. This work reveals that ZO gradient estimation exhibits *sample-wise alignment heterogeneity w.r.t. the global descent direction, and hard samples yield more informative ZO updates for minimizing the global objective than easy ones*. Based on this finding, we propose a difficulty-aware ZO (DaZO) algorithm that exploits sample-wise heterogeneity through sample reweighting. It down-weights easy samples while preserving informative hard samples. Further, we show that difficulty-aware reweighting reduces the global-gradient referenced estimation error and benefits faster convergence. Extensive experiments on RoBERTa-Large, Llama3-8B, and OPT models (1.3B/13B/30B) demonstrate significant improvement of DaZO over strong ZO baselines. Our work provides the first sample-centric view of ZO optimization, and shows that the quality of ZO gradient estimation depends systematically on the training samples. It is shown that difficulty-aware reweighting significantly improves ZO methods but has limited impact on FO methods, which highlights its particular relevance to the noisy gradient-estimation regime of ZO optimization. Code is available at `https://anonymous.4open.science/r/DaZO-E5A4/`.

1 INTRODUCTION

Large language models (LLMs) have revolutionized natural language processing, which have remarkable capabilities across a wide range of tasks. To adapt general-purpose LLMs to specific downstream tasks, fine-tuning typically remains an essential step. However, on resource-constrained devices, backpropagation (BP) fine-tuning is memory-prohibitive, as beyond the model parameters, BP must store gradients, optimizer states (e.g., momentum), and intermediate activations for the backward pass, which may largely exceed the memory budget of edge devices.

Zeroth-order (ZO) optimization has recently emerged as a memory-efficient alternative. ZO methods approximate gradients via finite differences using only forward pass evaluation, which avoids activation storage altogether. As a result, the memory cost of ZO fine-tuning can approach that of inference. Recent studies have successfully scaled ZO methods to LLMs with billions of parameters (Malladi et al., 2023), which enables fine-tuning on commodity GPUs where BP would be unaffordable.

Applying ZO optimization to LLM fine-tuning, a key challenge is the high variance of ZO gradient estimation, which grows linearly with the problem dimension (Duchi et al., 2015). Accordingly, many methods have been proposed to improve ZO fine-tuning (Chen et al., 2025; Guo et al., 2025; Gautam et al., 2024; Zhao et al., 2025b). Despite the progress, prior works predominantly focus on optimizer design, variance reduction, perturbation strategies. However, the *sample-centric* aspects of ZO optimization remain largely unexplored. In first-order training, it is well known that examples contribute unequally to learning, e.g., curriculum learning and hard example mining (Bengio et al., 2009; Kumar et al., 2010).

In this work, we take a sample-centric view of ZO fine-tuning and ask a natural but overlooked question: *Does the ZO gradient estimation interact with sample difficulty in a distinctive way that can*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

be exploited to improve optimization efficiency? Our key observation is that ZO gradient estimates exhibit pronounced *sample-wise alignment heterogeneity w.r.t. the global descent direction defined by the full gradient of the objective*. In particular, we find that ZO gradient estimates computed on harder samples exhibit stronger alignment with the global descent direction, and yield a higher directional signal-to-interference-plus-noise ratio (SINR), thus are more informative for minimizing the global objective. Motivated by this insight, we develop a difficulty-aware ZO (DaZO) algorithm that exploits this heterogeneity to improve ZO-based LLM fine-tuning. The main contributions are as follows:

- We reveal that, in ZO fine-tuning of LLMs, hard samples tend to yield more informative updates for minimizing the global objective, as quantified by a directional signal-to-interference-plus-noise ratio (SINR) that measures the component of the ZO gradient estimate aligned with the global descent direction.
- We propose DaZO, a difficulty-aware ZO algorithm that exploits sample-wise heterogeneity via sample reweighting. DaZO down-weights easy samples while preserving the contribution of more informative hard samples to improve ZO gradient estimation. Further, we show that the difficulty-aware reweighting reduces the global-gradient referenced estimation error.
- Extensive experiments on RoBERTa-Large, Llama3-8B, and OPT models (1.3B/13B/30B) demonstrate that DaZO outperforms strong ZO baselines and helps narrow the performance gap between ZO and first-order (FO) fine-tuning.

To the best of our knowledge, this work is the first to investigate and exploit sample-wise heterogeneity in ZO gradient estimation. Our results show that **difficulty-aware reweighting has little impact on FO fine-tuning (see Section 4.4), but significantly improves ZO fine-tuning**, which shows that our finding on sample-wise difficulty heterogeneity is particularly relevant to the noisy gradient-estimation regime of ZO optimization. A detailed related work is provided in Appendix A.

2 Sample-Wise Heterogeneity of ZO Gradient Estimation

2.1 Preliminary of ZO Optimization

ZO optimizers estimate gradients using only loss evaluation with forward passes. For a mini-batch $\mathcal{B}$, denote the empirical loss as

$$\mathcal{L}(\theta;\mathcal{B}) = \frac{1}{|\mathcal{B}|}\sum_{x\in\mathcal{B}} \mathcal{L}(\theta;x), \tag{1}$$

where $\theta \in \mathbb{R}^d$ collects the model parameters. ZO methods estimate gradients by probing $\mathcal{L}$ at randomly perturbed parameters. The classical two-point ZO gradient estimator is given by

$$\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}) = \frac{1}{q}\sum_{i=1}^{q} \frac{\mathcal{L}(\theta+\mu u_i;\mathcal{B}) - \mathcal{L}(\theta-\mu u_i;\mathcal{B})}{2\mu} u_i, \tag{2}$$

where $u_i \in \mathbb{R}^d$ is a random perturbation, e.g., $u_i \sim \mathcal{N}(0, I_d)$ or uniform on the unit sphere. $\mu > 0$ is the perturbation scale. q is the query number. Eq. (2) is also known as the q-query simultaneous perturbation stochastic approximation (q-SPSA) of gradient (SPALL, 1992) or randomized gradient estimator (RGE) (Nesterov & Spokoiny, 2017; Duchi et al., 2015). Similar to previous works (Malladi et al., 2023; Zhao et al., 2025b; Chen et al., 2025), we use $q = 1$ for efficiency.

Considering $q = 1$ and $u \sim \mathcal{N}(0, I_d)$, for $\mu \to 0$, it follows that $\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}) \to uu^T\nabla\mathcal{L}(\theta;\mathcal{B})$. Then, it can be seen that $\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})$ is an unbiased estimator of $\nabla\mathcal{L}(\theta;\mathcal{B})$ as $\mathbb{E}[\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})] = \mathbb{E}[uu^T\nabla\mathcal{L}(\theta;\mathcal{B})] = \nabla\mathcal{L}(\theta;\mathcal{B})$. With the SPSA gradient estimation, the model parameters can be updated via ZO-SGD as

$$\theta_{t+1} = \theta_t - \eta\widehat{\nabla}\mathcal{L}(\theta_t;\mathcal{B}_t), \tag{3}$$

where η is the learning rate, and $\mathcal{B}_t$ is the mini-batch at iteration t.

2.2 Sample Difficulty Scores

To study how sample difficulty affects ZO gradient estimation, we first introduce difficulty scores for individual training examples. Prior work has proposed various criteria, including predictive entropy

(Grandvalet & Bengio, 2004), margin-based uncertainty near decision boundaries (Scheffer et al., 2001), and gradient-dependent measures such as GraNd and EL2N (Paul et al., 2021). We consider simple *forward-only* difficulty metrics that can be computed without accessing gradients. For a sample (x, y), let the model output a categorical distribution $\hat{y} = p_\theta(\cdot|x) \in \mathbb{R}^L$ over L classes (the token vocabulary), and let $y \in \mathbb{R}^L$ denote the one-hot label distribution. We consider the following three scores.

Cross-Entropy Score (CES). We use the standard cross-entropy as a difficulty proxy: $\mathrm{CES}(x, y) = CE(\hat{y}, y)$. A larger CES indicates lower likelihood assigned to the correct label, hence a harder sample. In implementation, we normalize CES within each mini-batch to preserve relative difficulty while stabilizing scale across batches.

Prediction Margin Score (PMS). Let $i_{\max} = \arg\max_i p_\theta(i|x)$ denote the predicted class. The prediction margin is defined as the gap between the top-1 and top-2 predicted probabilities (Scheffer et al., 2001):

$$\mathrm{PMS}(x, y) = p_\theta(i_{\max}|x) - \max_{k \neq i_{\max}} p_\theta(k|x). \tag{4}$$

A smaller PMS indicates a more ambiguous prediction and thus higher difficulty.

Ground-Truth Margin Score (GMS). Let i_y be the index of the ground-truth label of y and $p_\theta(i_y|x)$ is the predicted probability of the correct class. We define the GMS as

$$\mathrm{GMS}(x, y) = p_\theta(i_y|x) - \max_{k \neq i_y} p_\theta(k|x). \tag{5}$$

Intuitively, GMS directly measures how strongly the model prefers the correct class over the most competitive alternative. A large GMS corresponds to an easy, confidently classified sample, while a small GMS indicates a hard sample near or across the decision boundary.

As shown in our experiments (Figure 12 in Appendix I), GMS yields the best performance. Therefore, we mainly use GMS in our analysis and experiments.

2.3 Alignment Heterogeneity of ZO Gradient Estimation with the Global Descent Direction

As the optimization target is the global objective over the entire dataset, we focus on how batch ZO estimation aligns with the full FO gradient $\nabla\mathcal{L}(\theta) := \mathbb{E}_{\mathcal{B}}\left[\nabla\mathcal{L}(\theta; \mathcal{B})\right]$, i.e., the global descent direction.

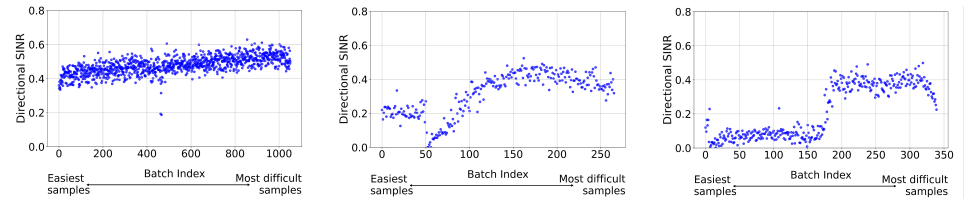
Importantly, we do not use the ZO estimation variance as a proxy for estimation quality. As shown in Figure 7 and the analysis in Appendix B, **harder samples typically induce larger FO gradient norm, and their batch-gradient referenced ZO estimation variance scales accordingly. Consequently, easy samples typically exhibit smaller batch-gradient referenced ZO variance simply because their underlying gradients are small.** This does not imply that their ZO gradients are more useful for minimizing the global objective. What matters is the signal along the global descent direction relative to the energy of the misaligned component and estimation noise.

Motivated by this, we adopt a relative direction-aware metric that measures the strength of the ZO update component aligned with the global descent direction, normalized by the residual energy orthogonal to that direction. This metric jointly captures *(i)* the alignment of the batch-induced gradient with the global descent direction, and *(ii)* the stochastic error introduced by ZO estimation, which enables a fair comparison across data batches of different difficulty.

Specifically, given the full gradient $\nabla\mathcal{L}(\theta)$, let $v_g = \frac{\nabla\mathcal{L}(\theta)}{\|\nabla\mathcal{L}(\theta)\|_2}$ denote the unit global descent direction induced by the full gradient. Then, we define the directional relative mean-squared error (MSE) of batch ZO gradient estimation w.r.t the full gradient as

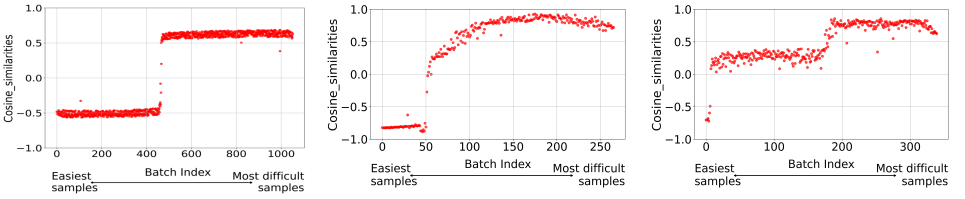
$$\mathrm{rMSE}(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})) = \frac{\mathbb{E}\left[\left\|\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}) - \langle\nabla\mathcal{L}(\theta;\mathcal{B}), v_g\rangle v_g\right\|_2^2\right]}{\langle\nabla\mathcal{L}(\theta;\mathcal{B}), v_g\rangle^2}. \tag{6}$$

This relative MSE (6) decomposes the ZO gradient estimate $\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})$ into a component parallel to the global descent direction v_g and a component orthogonal to v_g. The denominator $\langle\nabla\mathcal{L}(\theta;\mathcal{B}), v_g\rangle^2$ measures the squared magnitude of the *useful signal* that directly contributes to decreasing the global



(a) SST-2 (b) SST-5 (c) TREC

Figure 1: Directional SINR (8) (inverse of the directional relative MSE (6)) for samples of varying difficulty on the SST-2, SST-5 and TREC tasks using RoBERTa-Large, where samples are sorted by their GMS scores.



(a) SST-2 (b) SST-5 (c) TREC

Figure 2: Cosine similarity between batch FO gradient $\nabla\mathcal{L}(\theta;\mathcal{B})$ and the full gradient $\nabla\mathcal{L}(\theta)$ for samples of varying difficulty using RoBERTa-Large, where samples are sorted by their GMS scores.

objective. The numerator quantifies the energy of the misaligned (orthogonal) component, which has a noise plus interference (misalignment) decomposition as

$$\mathbb{E}\left[\left\|\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})-\langle\nabla\mathcal{L}(\theta;\mathcal{B}),v_g\rangle v_g\right\|_2^2\right]=\underbrace{\mathbb{E}\left[\left\|\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})-\nabla\mathcal{L}(\theta;\mathcal{B})\right\|_2^2\right]}_{\text{ZO estimation noise}} + \underbrace{\left\|\nabla\mathcal{L}(\theta;\mathcal{B})-\langle\nabla\mathcal{L}(\theta;\mathcal{B}),v_g\rangle v_g\right\|_2^2}_{\text{interference (misalignment)}}, \tag{7}$$

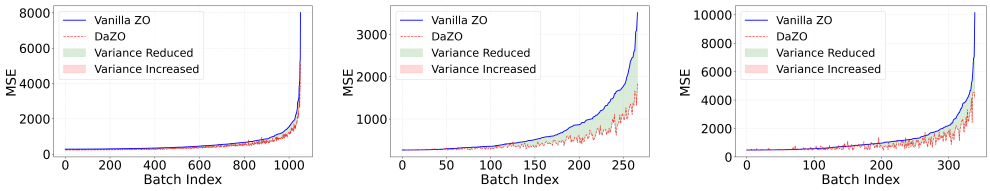
where the second term corresponds to the interference component of $\nabla\mathcal{L}(\theta;\mathcal{B})$ orthogonal to v_g. Thus, the inverse of the directional relative MSE (6) can be interpreted as directional signal-to-interference-plus-noise ratio (SINR)

$$\mathrm{SINR}(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}))=\frac{1}{\mathrm{rMSE}(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}))}. \tag{8}$$

This metric calculates the relative ratio of useful signal, and a larger SINR (equivalently a smaller rMSE) directly indicates that the ZO estimation is more informative for the global descent direction.

Figure 1 reports the directional SINR (8) (inverse of the rMSE (6)) for samples of varying difficulty on the SST-2, SST-5 and TREC tasks using RoBERTa-Large, where samples are sorted by their GMS scores. The MSE $\mathbb{E}\left[\left\|\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})-\langle\nabla\mathcal{L}(\theta;\mathcal{B}),v_g\rangle v_g\right\|_2^2\right]$ of each mini-batch is estimated over 100 independent perturbation sampling, and the ZO estimate is scaled by $\sqrt{d+2}$ to align the scale with that of the FO gradient, see Theorem B.2. We observe a clear trend, **hard samples have substantially higher SINR than easy samples.** This indicates that ZO gradient estimates computed on hard samples are more informative for the global descent direction of the training objective.

To further investigate the underlying mechanism of this trend, we examine in Figure 2 the cosine similarity between the batch FO gradient $\nabla\mathcal{L}(\theta;\mathcal{B})$ and the full FO gradient $\nabla\mathcal{L}(\theta)$ across different difficulty levels. The similarity consistently increases with sample difficulty, which implies that **FO gradients computed on hard samples exhibit stronger alignment with the full-gradient than those computed on easy samples.** Since $\mathbb{E}\{\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\}=\nabla\mathcal{L}(\theta;\mathcal{B})$ under the asymptotically unbiased condition with $\mu\to 0$, a similar alignment trend is expected to hold for the corresponding ZO gradient estimation $\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})$.



(a) SST-2 (b) SST-5 (c) TREC

Figure 3: Full-gradient referenced MSE of vanilla ZO (using $\mathcal{L}(\theta;\mathcal{B})$) and DaZO (using the reweighted loss $\mathcal{L}_w(\theta;\mathcal{B})$) using RoBERTa-Large. The MSE is computed w.r.t. the full gradient $\nabla\mathcal{L}(\theta)$. For visual clarity, batches are sorted by the MSE of vanilla ZO in ascending order.

Taken together, these analyses suggest that the global descent direction of the training objective is primarily driven by hard samples, whereas gradients induced by easy samples are weakly correlated with or may even deviate from this direction. As a consequence, ZO gradient estimates on hard samples provide more informative updates for minimizing the global loss.

3 The Proposed DaZO Method

3.1 Difficulty-Aware ZO Optimization Using Sample-Wise Reweighting

The above analysis shows that hard samples are more informative for optimizing the global objective, which motivates us to utilize difficulty-aware reweighting in ZO fine-tuning. To make each update to be as close as possible to the global descent direction, we re-weight samples within each mini-batch using a difficulty proxy to prioritize samples (hard samples) that are more likely to align with $\nabla\mathcal{L}(\theta)$ and suppressing those (easy samples) that contribute weakly aligned updates.

Specifically, DaZO uses a reweighted loss as

$$\mathcal{L}_w(\theta;\mathcal{B}) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} w_i\mathcal{L}(\theta;x_i,y_i), \tag{9}$$

where w_i is a weighting factor for each sample computed based on the difficulty of the sample (x_i, y_i). For the difficulty score GMS in Section 2.2 as an example, ω_i is computed as

$$w_i = \frac{1}{2}\left(1 - \mathrm{GMS}(x_i,y_i) + \lambda\right), \tag{10}$$

where $\lambda > 0$ is a small factor that ensures non-zero weighting for easy samples with $\mathrm{GMS}(x,y) \to 1$. In our experiments, the specific values of λ are detailed in Appendix H. The resulting weighted batch loss $\mathcal{L}_w(\theta;\mathcal{B})$ steers ZO updates toward directions that better match the global descent direction.

Note that, in implementation GMS and w_i are computed directly from one prediction of the two forward evaluations in the two-point estimator. Therefore, DaZO does not introduce any additional forward pass for computing the weights. The only additional computation for computing the margin and sample-wise loss reweighting is negligible compared with the forward passes.

Figure 3 compares the empirical full-gradient referenced MSE of the vanilla ZO estimator with that of DaZO. Specifically, we report $\frac{1}{M}\sum_{i=1}^{M}\left\|\frac{\widehat{\nabla}\mathcal{L}^{(i)}(\theta;\mathcal{B})}{\sqrt{d+2}} - \nabla\mathcal{L}(\theta)\right\|_2^2$ and $\frac{1}{M}\sum_{i=1}^{M}\left\|\frac{\widehat{\nabla}\mathcal{L}_w^{(i)}(\theta;\mathcal{B})}{\sqrt{d+2}} - \nabla\mathcal{L}(\theta)\right\|_2^2$ over $M = 100$ independent perturbation draws. The ZO estimates are scaled by $\sqrt{d+2}$ to align the scale of the ZO estimator with that of the FO gradient, see Theorem B.2. The results show that difficulty-aware reweighting consistently reduces the estimation error, which indicates that DaZO yields a less noisy and more informative ZO gradient estimate than vanilla ZO under the full-gradient reference.

3.2 Convergence Analysis

Assumption 3.1 (L-smoothness)**.** Given any batch data $\mathcal{B}$, the loss function $\mathcal{L}(\theta;\mathcal{B})$ is L-smooth with respect to all $\theta \in \mathbb{R}^d$.

Assumption 3.2 (full-gradient referenced MSE for DaZO)**.** The full-gradient referenced MSE for DaZO is bounded by ξ_d^2, i.e., $\mathbb{E}_{\mathcal{B}}\left[\left\|\frac{\widehat{\nabla}\mathcal{L}_w(\theta;\mathcal{B})}{\sqrt{d+2}} - \nabla\mathcal{L}(\theta)\right\|^2\right] \leq \xi_d^2$.

The smoothness condition on the loss function, stated in Assumption 3.1, is commonly adopted in the analysis of non-convex optimization and serves as a standard prerequisite for establishing convergence rates. Meanwhile, Assumption 3.2 relates to the empirical observation illustrated in Figure 3, namely that DaZO achieves a lower full-gradient referenced MSE compared to vanilla ZO. By examining the convergence rate in terms of ξ_d^2, we are able to quantitatively compare the convergence performance between DaZO and vanilla ZO.

Theorem 3.3 (Convergence rate of DaZO with SGD)**.** *Suppose Assumption 3.1 and 3.2 hold, and consider minimizing $\mathcal{L}$ using mini-batch gradients over T training steps, and learning rate $\eta = \frac{1}{2L\sqrt{T(d+2)}}$. By choosing $T > 1$, the convergence rate of the DaZO with SGD satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta_t)\|^2\right] \leq \mathcal{O}\left(\frac{1}{\sqrt{T}} + \xi_d^2\right).$$

The result in Theorem 3.3 demonstrates that DaZO achieves a faster convergence rate than vanilla ZO. To elaborate, assuming the full-gradient referenced MSE of vanilla ZO is bounded by σ_d^2, i.e., $\mathbb{E}_{\mathcal{B}}\left[\left\|\frac{\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})}{\sqrt{d+2}} - \nabla\mathcal{L}(\theta)\right\|^2\right] \leq \sigma_d^2$, the ratio between the convergence rates of DaZO and vanilla ZO scales as $O(\xi_d^2/\sigma_d^2)$. Since Figure 3 empirically shows $\xi_d^2 < \sigma_d^2$, we have $O(\xi_d^2/\sigma_d^2) < 1$, which implies a strictly faster convergence for DaZO.

4 Experiments

We evaluate our method on both a medium-scale masked language model and large-scale generative models. Specifically, we consider RoBERTa-Large (350M) (Liu et al., 2019) and four generative models, including OPT-1.3B/13B/30B (Zhang et al., 2022) and Llama3-8B (Grattafiori et al., 2024). We report results under standard LLM adaptation protocols, including zero-shot evaluation, linear probing (LP) (Alain & Bengio, 2016), and in-context learning (ICL) (Brown et al., 2020). We compare against strong ZO fine-tuning baselines, including MeZO (Malladi et al., 2023), ZO-Adam, HiZOO (Zhao et al., 2025b), LOZO (Chen et al., 2025), PseuZO (Yue et al., 2025), and PaZO (Zhao et al., 2025a). In addition to full-parameter ZO fine-tuning, we also evaluate PEFT methods including LoRA (Hu et al., 2022) and prefix-tuning (Li & Liang, 2021).

As shown in Figure 12 in Appendix I, using any of the three forward-only difficulty scores in Section 2.2, DaZO improves performance over the baseline, and GMS (5) yields the best result. Therefore, we use GMS in DaZO in all subsequent experiments. All experiments are conducted on NVIDIA A100 GPUs with 80GB memory. Detailed experiment settings are provided in Appendix H.

4.1 Masked Language Model

The first experiment considers RoBERTa-Large (350M) on six sentiment classification, natural language inference, and topic classification tasks. We investigate both few-shot and many-shot settings following (Gao et al., 2021). The few-shot setting uses $k=16$ examples per class, while the many-shot setting uses $k=512$ examples per class. The results for $k=16$ are presented in Table 1, while the results for $k=512$ are presented in Table 10 in Appendix I. The compared methods include zero-shot learning, LP, ZO-Adam, FO-Adam, MeZO, HiZOO, LOZO, and PaZO.

From Table 1, DaZO is competitive with recent strong ZO baselines (including HiZOO, LOZO, and PaZO) in PEFT, and achieves the best result in full fine-tuning. **In full fine-tuning, DaZO improves the average accuracy by 0.5% over the second-best method HiZOO, while using only half of its weight-memory overhead**. Across the six tasks, DaZO attains the best score on four. Notably, on SST-5 and MNLI, DaZO surpasses HiZOO by more than 2.6% and 2.8%, respectively.

4.2 Generative Language Models

Table 1: Results on RoBERTa-Large on six downstream tasks ($k = 16$). *LP*: linear probing. *FO-Adam*: full fine-tuning using first-order gradient optimization with Adam. We also evaluate PEFT methods (LoRA and prefix tuning) and report their best results. Results for DaZO are averaged over five runs. $(n \times M)$ denotes a memory overhead of n times the model weight memory M.

Task	SST-2	SST-5	SNLI	MNLI	RTE	TREC	Average
Zero-shot	79.0	35.5	50.2	48.8	51.4	32.0	49.5
LP	76.0	40.3	66.0	56.5	59.4	51.3	58.3
ZO-Adam $(3 \times M)$	90.3	38.8	69.1	59.2	61.4	52.4	61.7
FO-Adam $(6 \times M)$	92.9	48.5	77.5	70.0	67.1	85.0	73.5
MeZO (PEFT)	91.3	42.4	62.7	55.6	60.5	73.4	64.3
HiZOO (PEFT)	92.3	47.2	68.8	61.6	65.4	82.0	69.6
LOZO (PEFT)	88.0	42.9	74.0	62.7	60.2	82.2	68.3
PaZO (PEFT)	91.3	42.9	62.4	55.8	61.5	77.4	65.2
DaZO (PEFT)	88.0	46.3	70.8	64.6	65.6	77.3	68.8
MeZO	90.5 $\pm$1.2	45.5 $\pm$2.0	68.5 $\pm$3.9	58.7 $\pm$2.5	64.0 $\pm$3.3	76.9 $\pm$2.7	67.4
HiZOO $(2 \times M)$	92.3 $\pm$0.8	45.3 $\pm$1.1	74.6 $\pm$1.3	64.3 $\pm$1.7	66.8 $\pm$1.2	78.5 $\pm$1.3	70.3
LOZO	88.0 $\pm$5.8	41.3 $\pm$2.8	73.4 $\pm$4.0	61.6 $\pm$4.4	61.2 $\pm$5.3	77.9 $\pm$3.2	67.2
PaZO $(2 \times M)$	91.4 $\pm$0.8	44.6 $\pm$1.7	66.7 $\pm$2.6	56.4 $\pm$2.1	63.2 $\pm$5.2	70.8 $\pm$2.0	65.5
DaZO	92.4 $\pm$0.5	47.9 $\pm$1.9	71.7 $\pm$1.7	67.1 $\pm$2.2	65.6 $\pm$2.9	79.8 $\pm$1.3	70.8

Table 2: Results on OPT-1.3B using 1000 training samples. *ICL:* in-context learning. (↑) indicates the absolute improvement over the second-best result, evaluated under the same fine-tuning setting.

Method	SST-2	RTE	CB	BoolQ	WSC	WIC	MultiRC	Copa	ReCoRD	Average
Zero-shot	53.5	53.0	39.3	45.7	43.3	51.5	45.4	75.0	70.5	53.0
ICL	80.0	53.0	46.4	58.7	47.1	51.1	46.2	69.0	71.0	58.1
MeZO	88.5	66.1	67.8	63.4	57.6	60.6	56.3	76.0	72.3	67.6
MeZO (LoRA)	88.5	59.9	60.7	63.0	54.8	58.2	58.9	77.0	70.6	65.7
MeZO (prefix)	91.3	62.5	67.9	64.1	51.2	54.2	57.2	75.0	71.0	66.0
HiZOO	88.5	64.3	67.9	61.4	62.5	62.2	59.3	73.0	71.9	67.9
HiZOO (LoRA)	88.5	64.6	69.6	63.1	54.8	60.6	58.9	76.0	72.5	67.6
HiZOO (prefix)	91.3	63.2	67.9	63.6	57.7	53.8	54.5	75.0	70.9	66.4
PseuZO	90.7	63.3	75.0	67.0	57.0	59.7	60.6	76.0	60.9	67.8
PseuZO (LoRA)	91.2	58.8	79.0	65.8	51.0	51.8	54.3	83.0	60.2	66.1
PseuZO (prefix)	80.7	53.1	71.0	61.7	49.0	51.0	52.7	75.0	57.8	61.3
PaZO	89.0	66.4	69.6	63.4	61.5	63.2	57.6	75.0	72.1	68.6
PaZO (LoRA)	88.5	62.8	73.2	63.4	54.8	58.2	58.9	77.0	72.1	67.7
PaZO (prefix)	91.3	62.3	67.9	63.4	57.7	53.8	57.2	75.0	71.0	66.6
DaZO	92.6	68.4	80.0	68.0	61.5	64.2	62.1	85.0	68.9	72.3 (↑3.7)
DaZO (LoRA)	92.4	67.5	91.0	65.0	54.2	58.8	60.8	79.0	68.3	70.8 (↑3.1)
DaZO (prefix)	89.2	58.4	83.0	64.0	54.8	58.0	57.4	80.0	68.1	68.1 (↑1.5)

We further conduct experiments on four large-scale generative models, including OPT-1.3B/13B/30B and Llama3-8B. We consider various tasks covering classification and multiple-choice tasks. For each task, we randomly sample 1000, 500, and 1000 examples for training, validation, and test sets, respectively. The results are presented in Tables 2 to 5.

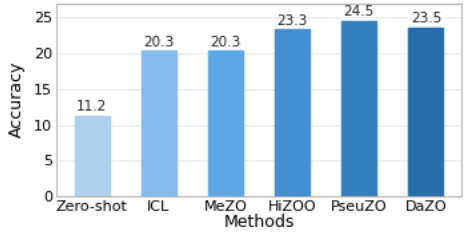


Figure 4: Results on the generation task DROP with OPT-1.3B using 1000 training samples.

On OPT-1.3B, the results of LoRA and prefix fine-tuning for each ZO method are also compared. As shown in Table 2, DaZO achieves the best overall performance on OPT-1.3B compared with MeZO, HiZOO, PseuZO and PaZO. Specifically, in terms of average accuracy, DaZO achieves the best performance that **improves over the second-best method PaZO by 3.7%, 3.1%, and 1.5% in the full-parameter, LoRA, and prefix tuning settings, respectively**. we also evaluate DaZO on the generation task DROP. The results are shown in Figure 4. It can be seen that DaZO achieves the second-best performance on the generation task, behind PseuZO.

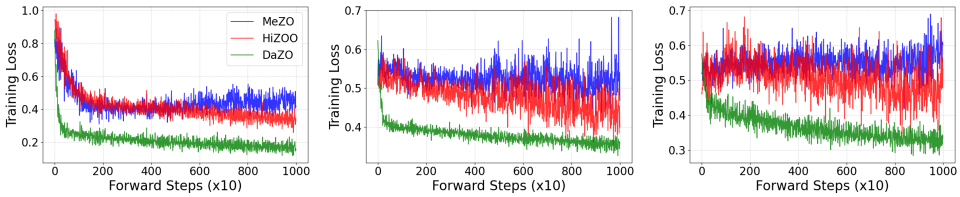
Furthermore, results on the Llama3-8B model again demonstrate the superiority of DaZO compared with the other ZO baselines including HiZOO and LOZO. As shown in Table 3, across the considered eight tasks, DaZO attains the best accuracy on six of them. Besides, in terms of average accuracy,

Table 3: Results on Llama3-8B using 1000 training samples. (↑) indicates the improvement over the second-best ZO method.

Method	SST-2	RTE	CB	BoolQ	WSC	WIC	MultiRC	Copa	Average
MeZO	92.2	74.4	69.6	76.4	63.5	57.8	77.6	88.0	74.9
HiZOO	93.5	75.1	69.6	76.8	63.5	59.7	78.2	89.0	75.7
LOZO	93.6	74.7	69.6	77.1	63.9	58.6	78.1	90.0	75.7
DaZO	94.4	73.6	80.0	79.6	65.4	59.9	76.0	94.0	77.9 (↑2.2)

Table 4: Results on OPT-13B using 1000 training samples. *ICL:* in-context learning. *LP*: linear probing. *FO-Adam*: full fine-tuning using first-order gradient optimization with Adam.

Method	SST-2	RTE	CB	BoolQ	WSC	WIC	MultiRC	Copa	ReCoRD	Average
Zero-shot	58.8	59.6	46.4	59.0	38.5	55.0	46.9	80.0	81.2	58.4
ICL	87.0	62.1	57.1	66.9	39.4	50.5	53.1	87.0	82.5	65.1
LP	93.4	68.6	67.9	59.3	63.5	60.2	63.5	55.0	27.1	62.1
MeZO	91.4	66.1	67.9	67.6	63.5	61.1	60.1	88.0	81.7	71.9
HiZOO	92.1	68.2	67.9	66.6	65.4	59.4	61.1	89.0	81.1	72.3
LOZO	91.7	70.4	69.6	71.9	63.5	60.8	63.0	89.0	81.3	73.5
DaZO	92.3	75.2	79.0	72.4	64.6	60.2	62.6	88.0	81.5	75.1 (↑1.6)



(a) Llama3-8B on SST-2 (b) Llama3-8B on RTE (c) Llama3-8B on BoolQ

Figure 5: Loss curves during training for different ZO methods with Llama3-8B.

DaZO achieves the **best performance with an average accuracy of 77.9% that improves over HiZOO and LOZO by a significant margin of 2.2%**.

From the results on the OPT-13B model across nine tasks, as shown in Table 4, DaZO again achieves the best performance compared with the ZO contenders. DaZO attains the **highest average accuracy of 75.1%, which is 1.6% higher than that of the second-best method LOZO (73.5%)**. On the RTE and CB tasks, DaZO significantly surpasses LOZO by 4.8% and 9.4%, respectively.

Finally, we evaluate DaZO on a larger model with 30B parameters. As shown in Table 5, on the OPT-30B model, DaZO also achieves the highest average accuracy, which **improves over the second-best method LOZO by 1.2%**.

Table 5: Results on OPT-30B using 1000 training samples.

Method	SST-2	RTE	WSC	WIC	Average
Zero-shot	56.7	52.0	38.5	50.2	49.4
ICL	81.9	66.8	56.7	51.3	64.2
MeZO	90.6	64.3	63.5	56.3	68.7
HiZOO	90.3	68.9	63.5	53.4	69.0
LOZO	92.8	65.3	64.4	57.2	69.9
DaZO	92.4	72.0	63.7	56.2	71.1 (↑1.2)

In addition to the accuracy, we also compare the convergence of loss curves during fine-tuning. Figure 5 compares the loss curves of MeZO, HiZOO and DaZO on the SST-2, RTE and BoolQ tasks with Llama3-8B. It can be seen that **DaZO shows significantly better convergence performance than MeZO and HiZOO**. This advantage can also be observed on other models, see the loss curves of OPT-1.3B shown in Figure 14 in Appendix I.

4.3 Memory Usage and Wall-clock Time Comparison

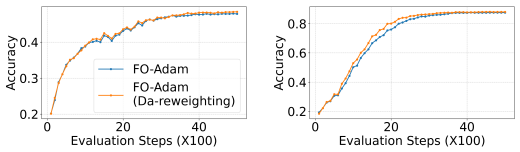
Memory usage. As shown in Figure 16 and Table 11 in Appendix I, DaZO and MeZO exhibit inference-level peak memory comparable to Zero-shot, since they do not store intermediate activations or any Hessian-related statistics. In contrast, HiZOO and PaZO maintain diagonal Hessian information, which roughly doubles the memory footprint under full fine-tuning. Even under PEFT

(e.g., prefix-tuning), FO-Adam remains much more memory-intensive than DaZO: on OPT-6.7B and OPT-13B, full fine-tuning with FO-Adam requires $11\times$ and $12\times$ higher peak memory, respectively.

Wall-clock time. DaZO does not introduce extra forward passes, as the sample weights are computed directly from one of the two forward evaluations in the two-point estimator. Table 12 in Appendix I compares the wall-clock time per step for MeZO, HiZOO, and DaZO across different model scales. DaZO achieves highly competitive time efficiency, which roughly matches that of MeZO.

4.4 Is Difficulty-Aware Reweighting Useful for FO Gradient Based Optimization?

We also evaluate the proposed difficulty-aware reweighting loss under the FO method with Adam, denoted as FO-Adam. Figure 6 compares standard FO-Adam with FO-Adam using the reweighted loss in (9) on RoBERTa-Large. The results show that the proposed difficulty-aware reweighting has little impact on FO-Adam.



(a) SST-5 (b) TREC

Figure 6: Accuracy curves of FO gradient based methods during training using RoBERTa-Large ($k = 16$). *FO-Adam*: vanilla FO gradient based Adam. *FO-Adam (Da-reweighting)*: FO gradient based Adam using difficulty-aware reweighting.

The key difference is that ZO methods rely on noisy gradient estimates, whereas FO methods use exact gradients. Difficulty-aware reweighting benefits ZO primarily by improving gradient-estimation quality, as it reduces the full-gradient-referenced estimation error. In contrast, FO methods do not suffer from such estimation noise, thus leaving much less room for improvement through this mechanism.

To further validate this, Table 6 compares MeZO and DaZO with different query numbers $q \in \{1, 10, 100\}$ under fixed forward-pass budgets. As q increases, the ZO estimator becomes less noisy. Table 6 shows a clear trend: the performance gain from difficulty-aware reweighting decreases as q grows. In the limit $q \to \infty$, the ZO estimator approaches the FO gradient, and the benefit of reweighting correspondingly diminishes. This suggests that DaZO is most beneficial when optimization is limited by ZO gradient-estimation noise.

Table 6: Results on SST-2 with RoBERTa-Large ($k = 16$) for $q \in \{1, 10, 100\}$ under fixed forward-pass (FP) budgets of 10k and 20k.

FP budget	Method	$q = 1$	$q = 10$	$q = 100$
10k	MeZO	69.8	64.5	51.2
	DaZO	87.1	73.1	56.3
20k	MeZO	78.1	75.5	72.7
	DaZO	92.4	89.1	84.4

Together with the above results, these findings indicate that **difficulty-aware reweighting substantially improves ZO fine-tuning methods, but has little impact on FO algorithms,** which implies that our finding on sample-wise difficulty heterogeneity is particularly relevant to the noisy gradient-estimation regime of ZO optimization.

Alation study on the parameter λ is provided in Appendix I.3.

5 Conclusion

We presented a novel data-centric perspective on ZO fine-tuning for LLMs. We uncovered the phenomenon of sample-wise alignment heterogeneity w.r.t. the global descent direction, and revealed that hard samples yield ZO gradient estimates that are better aligned with the global descent direction than easy ones. We proposed the DaZO algorithm, and showed that DaZO reduces the global-gradient referenced estimation error. Extensive experiments across RoBERTa-Large, Llama-3, and OPT models demonstrated the effectiveness of the proposed difficulty-aware reweighting method in comparison with existing strong baselines.

AI USE STATEMENT

Generative AI tools were used solely to assist with proofreading and polishing the manuscript, including grammar, wording, clarity, and consistency of terminology and mathematical notation. They were not used to conceive the research, develop the method, conduct experiments, or generate data or results. The authors reviewed and verified the suggested changes and take full responsibility for the final content of this work.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

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

Lejs Deen Behric, Liang Zhang, Bingcong Li, and Kiran Koshy Thekumparampil. Conmezo: Adaptive descent-direction sampling for gradient-free finetuning of large language models. *ArXiv*, 2025.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, pp. 41–48, 2009.

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

Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Konstantinos Parasyris, Jiancheng Liu, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. Deepzero: Scaling up zeroth-order optimization for deep model training. In *ICLR*, 2024.

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

Yiming Chen, Yuan Zhang, Liyuan Cao, Kun Yuan, and Zaiwen Wen. Enhancing zeroth-order finetuning for language models with low-rank structures. In *The Thirteenth International Conference on Learning Representations*, 2025.

Sizhe Dang, Yangyang Guo, Yanjun Zhao, Haishan Ye, Xiaodong Zheng, Guang Dai, and Ivor W Tsang. Fzoo: Fast zeroth-order optimizer for fine-tuning large language models towards adam-scale speed. *arXiv preprint arXiv:2506.09034*, 2025.

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.

Tianyu Gao, Adam Fisch, and Danqi 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*, 2021.

Tanmay Gautam, Youngsuk Park, Hao Zhou, Parameswaran Raman, and Wooseok Ha. Variance-reduced zeroth-order methods for fine-tuning language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 15180–15208, 2024.

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

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in Neural Information Processing Systems*, 17, 2004.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Wentao Guo, Jikai Long, Yimeng Zeng, Zirui Liu, Xinyu Yang, Yide Ran, Jacob R Gardner, Osbert Bastani, Christopher De Sa, Xiaodong Yu, Beidi Chen, and Zhaozhuo Xu. Zeroth-order fine-tuning of llms with transferable static sparsity. In *The Thirteenth International Conference on Learning Representations*, 2025.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

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

Shuoran Jiang, Qingcai Chen, Youcheng Pan, Yang Xiang, Yukang Lin, Xiangping Wu, Chuanyi Liu, and Xiaobao Song. Zo-adamu optimizer: Adapting perturbation by the momentum and uncertainty in zeroth-order optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18363–18371, 2024.

M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in Neural Information Processing Systems*, 23, 2010.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Wei Lin, Yining Jiang, Qingyu Song, Qiao Xiang, and Hong Xu. Agzo: Activation-guided zeroth-order optimization for llm fine-tuning. *arXiv preprint arXiv:2601.17261*, 2026.

Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

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

Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Sparse mezo: Less parameters for better performance in zeroth-order llm fine-tuning. *arXiv preprint arXiv:2402.15751*, 2024.

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*, 2025.

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.

Zhendong Mi, Qitao Tan, Grace Li Zhang, Zhaozhuo Xu, Geng Yuan, and Shaoyi Huang. Towards fast llm fine-tuning through zeroth-order optimization with projected gradient-aligned perturbations. *arXiv preprint arXiv:2510.18228*, 2025.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, 2021.

Yuezhang Peng, Yuxin Liu, Fei Wen, and Xie Chen. Muzo: Leveraging multiple queries and momentum for zeroth-order fine-tuning of large language models. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pp. 8577–8595, 2025.

Mingyue Qin, Shuyu Yin, Qinghai Guo, Peilin Liu, Xiaolin Huang, and Fei Wen. Zeroth-order forward-only snn training inspiring neuromorphic on-chip learning. In *Forty-third International Conference on Machine Learning*, 2026.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pp. 309–318, 2001.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235, 2020.

JC SPALL. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

Yan Sun, Tiansheng Huang, Liang Ding, Li Shen, and Dacheng Tao. Tezo: Empowering the low-rankness on the temporal dimension in the zeroth-order optimization for fine-tuning llms. *arXiv preprint arXiv:2501.19057*, 2025.

Ziming Yu, Pan Zhou, Sike Wang, Jia Li, Mi Tian, and Hua Huang. Zeroth-order fine-tuning of llms in random subspaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4475–4485, 2025.

Pengyun Yue, Xuanlin Yang, Mingqing Xiao, and Zhouchen Lin. Pseuzo: Pseudo-zeroth-order algorithm for training deep neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Kairun Zhang, Haoyu Li, Yanjun Zhao, Yifan Sun, and Huan Zhang. Learning a zeroth-order optimizer for fine-tuning llms. *arXiv preprint arXiv:2510.00419*, 2025.

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

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark. In *International Conference on Machine Learning*, 2024.

Hanzhen Zhao, Shihong Ding, Cong Fang, and Zhouchen Lin. Pazo: Preconditioned accelerated zeroth-order optimization for fine-tuning llms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2025a.

Yanjun Zhao, Sizhe Dang, Haishan Ye, Guang Dai, Yi Qian, and Ivor W Tsang. Second-order fine-tuning without pain for llms: A hessian informed zeroth-order optimizer. In *The Thirteenth International Conference on Learning Representations*, 2025b.

The contents of Appendix are as follows:

- Section A presents a detailed Related Work, including fine-tuning methods of LLMs, ZO optimization, and ZO optimization for LLM fine-tuning.
- Section B presents an analysis on batch-gradient referenced variance of ZO gradient estimation.
- Section C provides more empirical results on the alignment between FO batch-gradient with FO full-gradient.
- Section D provides proof of Theorem 3.3.
- Section E provides proof of Theorem B.1.
- Section F provides proof of Theorem B.2.
- Section G presents more results to demonstrate the estimation MSE reduction of the proposed difficulty-aware reweighting.
- Section H presents experiment setup, including datasets and hyperparameter settings.
- Section I presents more experiment results, ablation study, loss curve comparison, memory usage comparison and wall-clock time comparison.

A Related Work

A.1 Fine-Tuning of LLMs

Traditional optimization for LLMs is dominated by first-order optimizers. For LLM adaptation, parameter efficient fine-tuning (PEFT) techniques have attracted considerable attention, such as low-rank fine-tuning (Hu et al., 2022), adapter based methods (Houlsby et al., 2019), prompt tuning (Shin et al., 2020; Gao et al., 2021), and prefix tuning (Li & Liang, 2021). While these PEFT methods largely reduce the trainable parameters, they still incur substantial memory overhead from storing activations for BP.

A.2 Zeroth-Order Optimization

ZO optimization uses only function evaluations to estimate gradient, and hence is attractive when gradients are unavailable or expensive to compute. ZO methods have attracted much attention in modern machine learning (ML) (Chen et al., 2019; Liu et al., 2020; Qin et al., 2026). ZO optimization typically has a dimension-dependent convergence guarantees (Nesterov & Spokoiny, 2017; Duchi et al., 2015; Ghadimi & Lan, 2013). To address the bottleneck that estimation variance scales linearly with dimension, various methods have been proposed, such as variance reduction (Liu et al., 2018; Ji et al., 2019) and sparse update (Balasubramanian & Ghadimi, 2018; Chen et al., 2024).

A.3 Zeroth-Order Optimization for LLM Fine-Tuning

The seminal work, MeZO (Malladi et al., 2023), demonstrates that LLMs can be fine-tuned with inference-level memory budget using forward-only ZO gradient estimator. Then, many methods have been developed to mitigate the high variance and slow convergence inherent in ZO methods (Zhang et al., 2024).

Low-rank update methods: Exploiting the low-rank structure of the parameter updates is very useful to address the high-variance challenge posed by the high dimensionality of LLMs. LOZO (Chen et al., 2025) designs a ZO gradient estimator to effectively capture the low-rank gradient structure in LLM fine-tuning, whilst TeZO (Sun et al., 2025) considers jointly capturing the low-rankness across both the model and temporal dimension. Moreover, SubZero (Yu et al., 2025) introduces a low-rank perturbation approach using a random subspace, and P-GAP (Mi et al., 2025) adopts projected gradient-aligned perturbations (Ma & Huang, 2025) in low-dimensional gradient space. AGZO (Lin et al., 2026) further considers activation informed low-rank perturbation.

Sparse update methods: Updating only a small set of parameters can largely reduce the dimension of the optimization space and hence substantially reduce the variance of ZO gradient estimation. (Guo et al., 2025) only fine-tunes a static subset (about 0.1%) of "sensitive" parameters identified during pre-training. Sparse-MeZO (Liu et al., 2024) applies ZO update only to a subset of parameters chosen based on their weight magnitude.

Variance reduction methods: MeZO-SVRG (Gautam et al., 2024) couples ZO methods with variance reduction techniques to enhance stability and convergence for LLM fine-tuning.

Momentum methods: (Jiang et al., 2024) adapts perturbation based on the momentum and uncertainty to stabilize the convergence, whilst (Peng et al., 2025) combines multiple queries and momentum for variance reduction.

Curvature-aware methods: (Zhao et al., 2025b) introduces a Hessian informed ZO optimizer HiZOO to enhance ZO fine-tuning of LLMs. Subsequently, (Zhao et al., 2025a) provides analysis on Hessian preconditioned ZO optimization. Moreover, FZOO (Dang et al., 2025) utilizes multiple queries for each batch to probe the loss curvature for adaptive weighting of the learning rate, whilst (Zhang et al., 2025) leverages the block-wise structure of Hessian to learns an adaptive per-block variance.

In addition, (Yue et al., 2025) proposes a pseudo ZO method that utilizes the differentiation of model outputs to compute a stochastic Jacobian estimator. Moreover, (Behric et al., 2025) utilizes adaptive directional sampling to improve the perturbation efficiency.

B Analysis on Batch-Gradient Referenced Variance of ZO Gradient Estimation

The analysis in Section 2.3 focuses on the alignment of ZO gradient estimation with the full-gradient, where deviations are measured with respect to the first-order (FO) gradient computed over the entire dataset, i.e., with respect to the global descent direction. Here, we consider the batch-gradient referenced variance of ZO gradient estimation, where deviations are measured with respect to the FO gradient of the corresponding mini-batch, i.e., $\nabla\mathcal{L}(\theta;\mathcal{B})$.

First, we show that the difficulty score GMS is related to the gradient norm.

Theorem B.1 (Relation between GMS and gradient norm)**.** *Given a neural network model with parameter θ, a training data pair (x, y), we denote $z \in \mathbb{R}^L$ as the output of the neural network $f(\theta; x)$ before goes to softmax, and the corresponding cross-entropy loss as $\mathcal{L}(\theta; x)$, the gradient norm $\|\nabla\mathcal{L}(\theta; x)\|_2$ satisfies:*

$$\frac{1}{2}\sqrt{\frac{n}{n-1}}\lambda_{\min}(1-\mathrm{GMS}(x,y)) \le \|\nabla\mathcal{L}(\theta;x)\|_2 \le \sqrt{2}\lambda_{\max}(1-\mathrm{GMS}(x,y)), \tag{11}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalue of the Jacobian matrix $\frac{\partial z}{\partial\theta}$.

Next, we analyze how the batch-gradient referenced variance and a relative variance metric of ZO gradient estimation vary across sample difficulty levels. For the ZO estimation $\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})$ of the first-order gradient $\nabla\mathcal{L}(\theta;\mathcal{B})$ on a mini-batch $\mathcal{B}$ by (2), we define the *batch-gradient referenced variance* as

$$\mathrm{Var}_{\mathcal{B}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right) = \mathbb{E}_u\left[\left\|\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}) - \mathbb{E}_u\left[\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right]\right\|^2\right] \tag{12}$$

Note that, since the two-point estimator (2) is asymptotically unbiased as $\mu \to 0$, i.e., $\lim_{\mu\to 0}\mathbb{E}_u\left[\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right] = \nabla\mathcal{L}(\theta;\mathcal{B})$, the variance (12) is equivalent to the mean-squared error (MSE) of the estimator.

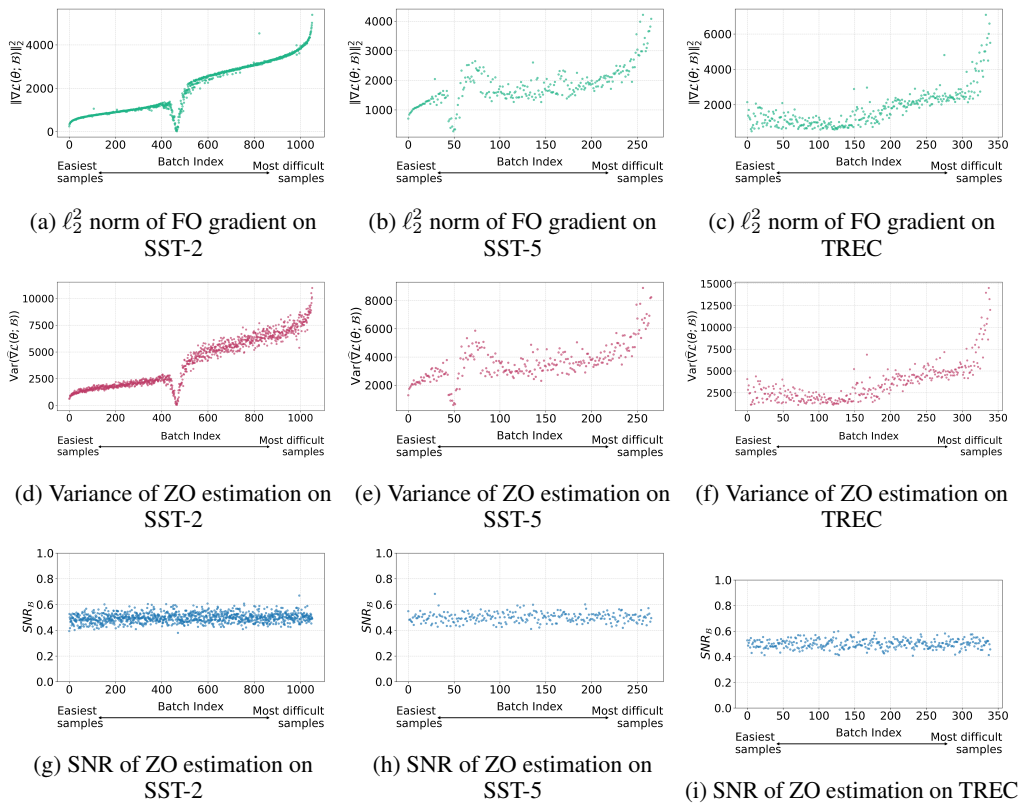
Moreover, since the norm of the FO gradient varies with sample difficulty, we introduce a *relative variance* metric for ZO gradient estimation, defined as

$$\mathrm{rVar}_{\mathcal{B}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right) = \frac{\mathrm{Var}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right)}{\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2}, \tag{13}$$

whose inverse can be interpreted as the *estimator signal-to-noise ratio (SNR)* under asymptotic unbiasedness

$$\mathrm{SNR}_{\mathcal{B}} = \frac{1}{\mathrm{rVar}_{\mathcal{B}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right)}. \tag{14}$$

Now, we show that the ZO estimation variance is proportional to $\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2$, while the relative variance is a constant that depends only on d and q.



(a) ℓ_2^2 norm of FO gradient on SST-2 (b) ℓ_2^2 norm of FO gradient on SST-5 (c) ℓ_2^2 norm of FO gradient on TREC

(d) Variance of ZO estimation on SST-2 (e) Variance of ZO estimation on SST-5 (f) Variance of ZO estimation on TREC

(g) SNR of ZO estimation on SST-2 (h) SNR of ZO estimation on SST-5 (i) SNR of ZO estimation on TREC

Figure 7: Variance and SNR of ZO gradient estimation across samples of varying difficulty on the SST-2, SST-5 and TREC tasks with RoBERTa-Large, where samples are sorted by their **GMS**. *Row 1*: Squared ℓ_2 norm of first-order (FO) gradient, $\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2$. *Row 2*: Variance of ZO gradient estimation, computed by $\widehat{\mathrm{Var}}(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}))$ in (18), over $M = 100$ independent runs. *Row 3*: SNR (14) computed by $\frac{\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2}{\widehat{\mathrm{Var}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right)}$.

Theorem B.2 (Relation between batch-gradient referenced ZO estimation variance and FO gradient norm)**.** *Assume that $\mathcal{L}$ is twice continuously differentiable with bounded Hessian, i.e., $\mathcal{L} \in C^2$ and $\|\nabla^2\mathcal{L}(\theta;\mathcal{B})\|_{\mathrm{op}} \leq L$. For Gaussian perturbations $u_i \sim \mathcal{N}(0, I_d)$ and $\mu \to 0$, the two-point ZO gradient estimator* (2) *satisfies*

$$\lim_{\mu\to 0} \mathbb{E}_u \left[\left\| \widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}) \right\|^2 \right] = \left(1 + \frac{d+1}{q}\right) \|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2, \tag{15}$$

and the variance (12) *satisfies*

$$\lim_{\mu\to 0} \mathrm{Var}_{\mathcal{B}} \left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}) \right) = \frac{d+1}{q} \|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2 . \tag{16}$$

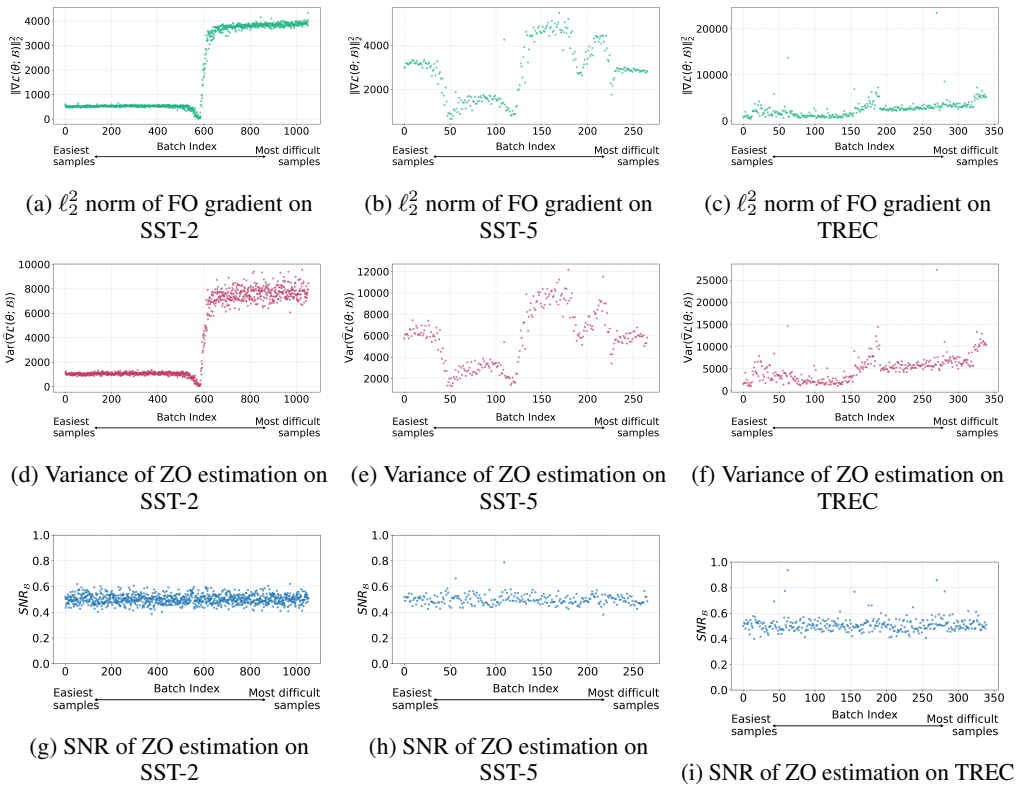
Furthermore, the relative variance (13) *satisfies*

$$\lim_{\mu\to 0} \mathrm{rVar}_{\mathcal{B}}\big(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\big) = \frac{d+1}{q}, \tag{17}$$

which is independent of $\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2$.

Remark 1. *Theorem B.2 shows that the variance of the ZO gradient estimator scales proportionally to the squared ℓ_2-norm of the corresponding FO gradient, $\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2$, whereas the relative variance (also the SNR* (14)*) is independent of $\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2$. Combined with Theorem B.1, which links gradient norm to sample difficulty, this yields two implications:*

***(i) The ZO estimation variance is expected to increase for harder samples**.*



(a) ℓ_2^2 norm of FO gradient on SST-2 (b) ℓ_2^2 norm of FO gradient on SST-5 (c) ℓ_2^2 norm of FO gradient on TREC

(d) Variance of ZO estimation on SST-2 (e) Variance of ZO estimation on SST-5 (f) Variance of ZO estimation on TREC

(g) SNR of ZO estimation on SST-2 (h) SNR of ZO estimation on SST-5 (i) SNR of ZO estimation on TREC

Figure 8: Variance and SNR of ZO gradient estimation across samples of varying difficulty on the SST-2, SST-5 and TREC tasks with RoBERTa-Large, where samples are sorted by their **CES**. *Row 1*: Squared ℓ_2 norm of first-order gradient, $\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2$. *Row 2*: Variance of ZO gradient estimation, computed by $\widehat{\mathrm{Var}}(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}))$ in (18), over $M = 100$ independent runs. *Row 3*: SNR (14) computed by $\frac{\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2}{\widehat{\mathrm{Var}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right)}$.

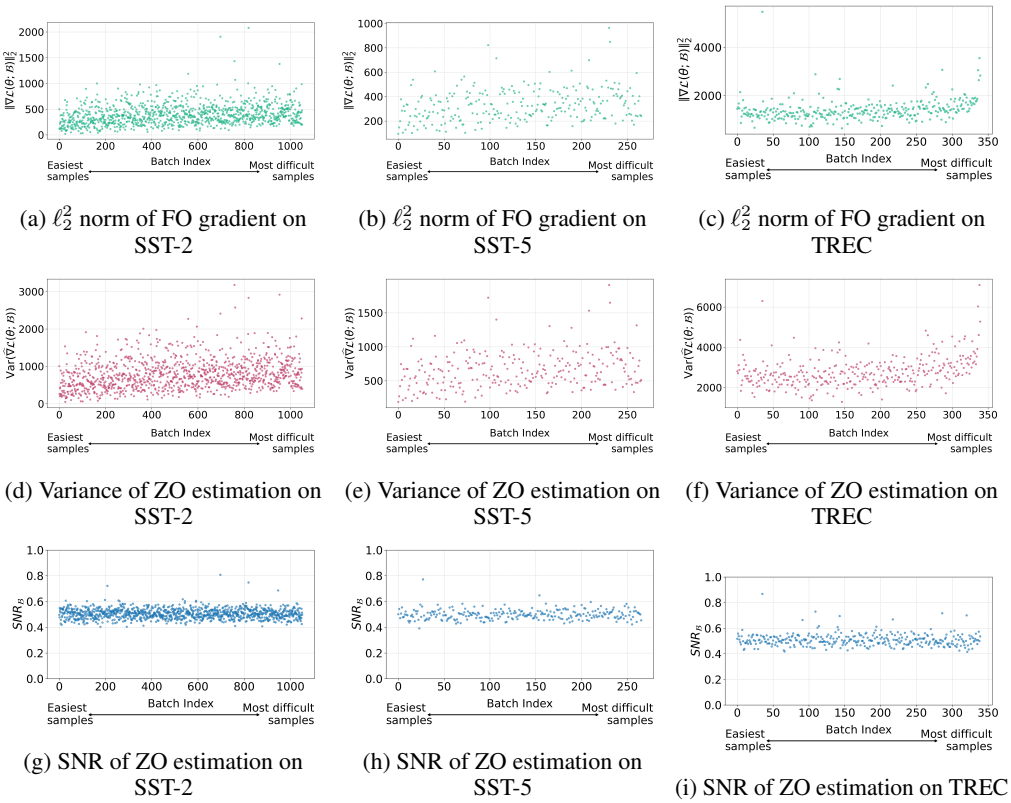
(ii) The relative variance and SNR should remain approximately constant across difficulty levels.

Next, we empirically examine these implications through controlled experiments. We estimate the variance (12) using M independent perturbation samples as

$$\widehat{\mathrm{Var}}_{\mathcal{B}}\big(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\big) = \frac{1}{M}\sum_{i=1}^{M}\left\|\frac{\widehat{\nabla}\mathcal{L}^{(i)}(\theta;\mathcal{B})}{\sqrt{d+2}} - \nabla\mathcal{L}(\theta;\mathcal{B})\right\|_2^2, \tag{18}$$

where d is the parameter dimension and $\widehat{\nabla}\mathcal{L}^{(i)}(\theta;\mathcal{B})$ is the i-th independent ZO gradient estimate on the mini-batch $\mathcal{B}$. The normalization factor $\sqrt{d+2}$ is introduced to align the scale of the ZO estimator with that of the true gradient, which is motivated by (15) in Theorem B.2. As a result, dividing by $\sqrt{d+2}$ yields a magnitude comparable to that of the true first-order gradient $\nabla\mathcal{L}(\theta;\mathcal{B})$ when $q = 1$, which enables a meaningful comparison of the signal and noise across samples.

Figure 7 presents the empirical variance $\widehat{\mathrm{Var}}(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}))$ in (18) and the corresponding SNR computed by $\frac{\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2}{\widehat{\mathrm{Var}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right)}$ (inverse relative variance) of ZO gradient estimates for samples of varying difficulty on SST-2, SST-5, and TREC using RoBERTa-Large, where samples are sorted by their GMS scores. We observe a clear trend: **the variance is proportional to the squared ℓ_2-norm of the FO gradient, and both quantities tend to increase with sample difficulty, whereas the SNR (also the relative**



(a) ℓ_2^2 norm of FO gradient on SST-2

(b) ℓ_2^2 norm of FO gradient on SST-5

(c) ℓ_2^2 norm of FO gradient on TREC

(d) Variance of ZO estimation on SST-2

(e) Variance of ZO estimation on SST-5

(f) Variance of ZO estimation on TREC

(g) SNR of ZO estimation on SST-2

(h) SNR of ZO estimation on SST-5

(i) SNR of ZO estimation on TREC

Figure 9: Variance and SNR of ZO gradient estimation across samples of varying difficulty on the SST-2, SST-5 and TREC tasks with RoBERTa-Large, where samples are sorted by their **PMS**. *Row 1*: Squared ℓ_2 norm of first-order gradient, $\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2$. *Row 2*: Variance of ZO gradient estimation, computed by $\widehat{\mathrm{Var}}(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}))$ in (18), over $M = 100$ independent runs. *Row 3*: SNR (14) computed by $\frac{\|\nabla\mathcal{L}(\theta;\mathcal{B})\|_2^2}{\widehat{\mathrm{Var}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right)}$.

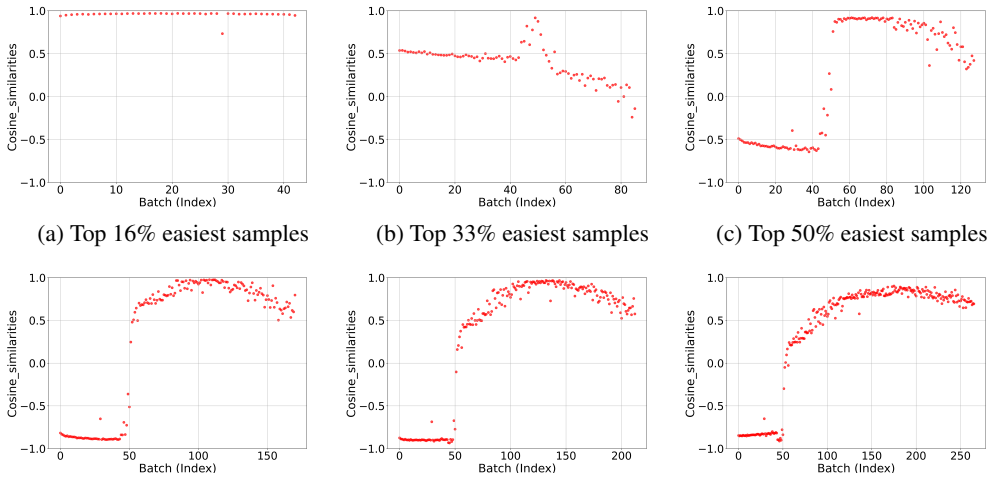
variance) remains relatively constant across difficulty levels. These results accord well with the theoretical results.

The results for the other two difficulty scores CES and PMS (4) are provided in Figure 8 and Figure 9, respectively. In our experiments, GMS yields better performance than CES and PMS, see Figure 12, which indicates that GMS is a better proxy of sample difficulty.

C More Results on the Alignment between FO Batch-Gradient with FO Full-Gradient

In Section 2.3, we investigated the cosine similarity between the batch FO gradient $\nabla\mathcal{L}(\theta;\mathcal{B})$ and the full FO gradient $\nabla\mathcal{L}(\theta)$ across different difficulty levels, see Figure 2. The results show that the similarity consistently increases with sample difficulty, which implies that FO gradients computed on hard samples exhibit stronger alignment with the full-gradient than those computed on easy samples.

Here, we further investigate the trend, we examine in Figure 10 the cosine similarity between the batch FO gradient $\nabla\mathcal{L}(\theta;\mathcal{B})$ and the full FO gradient $\nabla\mathcal{L}(\theta)$ across different difficulty levels. We first sort all the samples by their GMS scores. Then, we sample top-r easiest samples to compute the full gradient $\nabla\mathcal{L}(\theta)$, for $r \in \{16\%, 33\%, 50\%, 66\%, 83\%, 100\%\}$, and subsequently compute the cosine similarity of the batch ZO gradients in these top 16% easiest samples w.r.t. $\nabla\mathcal{L}(\theta)$. It can been seen that, as the batches used to compute the full gradient increase, the similarity between the easy samples and the full gradient consistently decreases, which implies that **FO gradients computed on**



(a) Top 16% easiest samples (b) Top 33% easiest samples (c) Top 50% easiest samples

(d) Top 66% easiest samples (e) Top 83% easiest samples (f) All samples in SST-5 task

Figure 10: Cosine similarity between batch FO gradient $\nabla\mathcal{L}(\theta;\mathcal{B})$ and the full gradient $\nabla\mathcal{L}(\theta)$ for samples of varying difficulty on the SST-5 task using RoBERTa-Large. We first sort all the samples by their GMS scores. Then, we sample top-r easiest samples to compute the full gradient $\nabla\mathcal{L}(\theta)$, for $r \in \{16\%, 33\%, 50\%, 66\%, 83\%, 100\%\}$. For example, in (a) the full gradient $\nabla\mathcal{L}(\theta)$ is computed using the top 16% easiest samples, and then the cosine similarity of the batch ZO gradients in these top 16% easiest samples are compared w.r.t. $\nabla\mathcal{L}(\theta)$.

hard samples exhibit stronger alignment with the full-gradient than those computed on easy samples.

D Proof of Theorem 3.3

Proof of Theorem 3.3. Combining L-smooth property of $\mathcal{L}(\theta)$ given by Assumption 3.1 with the parameter update rule of DaZO under a mini-batch $\mathcal{B}_t$ and take the expectation on $\mathcal{B}_t$, we have

$$\begin{aligned}\mathbb{E}_{\mathcal{B}_t}[\mathcal{L}(\theta_{t+1})] &\leq \mathcal{L}(\theta_t) - \eta\nabla\mathcal{L}(\theta_t)^\top\mathbb{E}_{\mathcal{B}_t}\left[\widehat{\nabla}\mathcal{L}_w(\theta_t;\mathcal{B}_t)\right] + \frac{L}{2}\mathbb{E}_{\mathcal{B}_t}\left[\left\|\eta\widehat{\nabla}\mathcal{L}_w(\theta_t;\mathcal{B}_t)\right\|^2\right] \\ &= \mathcal{L}(\theta_t) - \eta_d\nabla\mathcal{L}(\theta_t)^\top\mathbb{E}_{\mathcal{B}_t}\left[\frac{\widehat{\nabla}\mathcal{L}_w(\theta_t;\mathcal{B}_t)}{\sqrt{d+2}}\right] + \frac{L\eta_d^2}{2}\mathbb{E}_{\mathcal{B}_t}\left[\left\|\frac{\widehat{\nabla}\mathcal{L}_w(\theta_t;\mathcal{B}_t)}{\sqrt{d+2}}\right\|^2\right],\end{aligned} \tag{19}$$

where η represents the learning rate, and $\eta = \frac{\eta_d}{\sqrt{d+2}}$.

For the second of the RHS of (19), we have

$$\begin{aligned}&\nabla\mathcal{L}(\theta_t)^\top\mathbb{E}_{\mathcal{B}_t}\left[\frac{\widehat{\nabla}\mathcal{L}_w(\theta_t;\mathcal{B}_t)}{\sqrt{d+2}}\right] \\ &\overset{(a)}{\geq}\frac{1}{2}\left\|\nabla\mathcal{L}(\theta_t)\right\|^2 - \frac{1}{2}\left\|\nabla\mathcal{L}(\theta_t) - \mathbb{E}_{\mathcal{B}_t}\left[\frac{\widehat{\nabla}\mathcal{L}_w(\theta_t;\mathcal{B}_t)}{\sqrt{d+2}}\right]\right\|^2 \\ &\overset{(b)}{\geq}\frac{1}{2}\left\|\nabla\mathcal{L}(\theta_t)\right\|^2 - \frac{1}{2}\mathbb{E}_{\mathcal{B}_t}\left[\left\|\nabla\mathcal{L}(\theta_t) - \frac{\widehat{\nabla}\mathcal{L}_w(\theta_t;\mathcal{B}_t)}{\sqrt{d+2}}\right\|^2\right] \\ &\overset{(c)}{\geq}\frac{1}{2}\left\|\nabla\mathcal{L}(\theta_t)\right\|^2 - \frac{1}{2}\xi_d^2,\end{aligned} \tag{20}$$

where inequality (a) holds because $a^\top b \geq \frac{1}{2}\|a\|^2 - \frac{1}{2}\|a-b\|^2$ holds for any two vectors $a, b \in \mathbb{R}^d$, inequality (b) holds because of Jensen's inequality, and inequality (c) holds because of Assumption

3.2. Besides, by Assumption 3.2 and Cauchy-Schwarz inequality, we have

$$\mathbb{E}_{\mathcal{B}_t}\left[\left\|\frac{\widehat{\nabla}\mathcal{L}_w(\theta_t;\mathcal{B}_t)}{\sqrt{d+2}}\right\|^2\right] \leq 2\mathbb{E}_{\mathcal{B}_t}\left[\left\|\frac{\widehat{\nabla}\mathcal{L}_w(\theta_t;\mathcal{B}_t)}{\sqrt{d+2}} - \nabla\mathcal{L}(\theta_t)\right\|^2\right] + 2\left\|\nabla\mathcal{L}(\theta_t)\right\|^2 \leq 2\left\|\nabla\mathcal{L}(\theta_t)\right\|^2 + 2\xi_d^2. \tag{21}$$

Combining (19), (20) and (21), we have

$$\begin{aligned}\mathbb{E}_{\mathcal{B}_t}[\mathcal{L}(\theta_{t+1})] &\leq \mathcal{L}(\theta_t) - \frac{\eta_d}{2}\|\nabla\mathcal{L}(\theta_t)\|^2 + \frac{\eta_d}{2}\xi^2 + L\eta_d^2\left(\|\nabla\mathcal{L}(\theta_t)\|^2 + \xi^2\right)\\ &= \mathcal{L}(\theta_t) - \left(\frac{\eta_d}{2} - L\eta_d^2\right)\|\nabla\mathcal{L}(\theta_t)\|^2 + \left(\frac{\eta_d}{2} + L\eta_d^2\right)\xi_d^2.\end{aligned}$$

We re-arrange the above inequality to get the gradient norm on the left side

$$\left(\frac{\eta_d}{2} - L\eta_d^2\right)\|\nabla\mathcal{L}(\theta_t)\|^2 \leq \mathcal{L}(\theta_t) - \mathbb{E}_{\mathcal{B}_t}\left[\mathcal{L}(\theta_{t+1})\right] + \left(\frac{\eta_d}{2} + L\eta_d^2\right)\xi_d^2.$$

Sum the above inequality from $t=0$ to $T-1$ with iterative expectation on $\mathcal{B}_0, ..., \mathcal{B}_{t-1}$, then we have

$$\begin{aligned}&\sum_{t=0}^{T-1}\left(\frac{\eta_d}{2} - L\eta_d^2\right)\mathbb{E}_{\mathcal{B}_0,...,\mathcal{B}_{t-1}}\left[\|\nabla\mathcal{L}(\theta_t)\|^2\right]\\ \leq &\sum_{t=0}^{T-1}\left[\mathbb{E}_{\mathcal{B}_0,...,\mathcal{B}_{t-1}}\left[\mathcal{L}(\theta_t)\right] - \mathbb{E}_{\mathcal{B}_0,...,\mathcal{B}_t}\left[\mathcal{L}(\theta_{t+1})\right]\right] + T\left(\frac{\eta_d}{2} + L\eta_d^2\right)\xi^2\\ \leq &\mathcal{L}(\theta_0) - \mathbb{E}_{\mathcal{B}_0,...,\mathcal{B}_{T-1}}\left[\mathcal{L}(\theta_T)\right] + T\left(\frac{\eta_d}{2} + L\eta_d^2\right)\xi_d^2.\end{aligned}$$

Using $\mathcal{L}(\theta^*)$ to replace $\mathbb{E}_{\mathcal{B}_0,...,\mathcal{B}_{T-1}}[\mathcal{L}(\theta_T)]$ because of $\mathcal{L}(\theta^*) \leq \mathbb{E}_{\mathcal{B}_0,...,\mathcal{B}_{T-1}}[\mathcal{L}(\theta_T)]$, and divided both sides with $\sum_{t=0}^{T-1}(\frac{\eta_d}{2} - L\eta_d^2) = \frac{\sqrt{T}-1}{4L}$ where $\eta_d = \frac{1}{2L\sqrt{T}}$ and $T > 1$, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}_{\mathcal{B}_0,...,\mathcal{B}_{t-1}}[\|\nabla\mathcal{L}(\theta_t)\|^2] \leq \frac{4L[\mathcal{L}(\theta_0) - \mathcal{L}(\theta^*)]}{\sqrt{T}-1} + \left(1 + \frac{2}{\sqrt{T}-1}\right)\xi_d^2.$$

The convergence rate can be represented as the leading term, which is $\mathcal{O}\left(\frac{1}{\sqrt{T}} + \xi_d^2\right)$. The proof is complete. □

E Proof of Theorem B.1

Proof. We first show the gradient norm of $\frac{\partial\mathcal{L}}{\partial z}$ satisfies

$$\frac{\sqrt{n/(n-1)}}{2}(1 - \mathrm{GMS}(x,y)) \leq \left\|\frac{\partial\mathcal{L}(\theta;x)}{\partial z}\right\|_2 \leq \sqrt{2}(1 - \mathrm{GMS}(x,y)). \tag{22}$$

Because the loss function is a cross entropy loss, then for any z_i, which is the i-th element of z, we have

$$\frac{\partial\mathcal{L}}{\partial z_i} = \frac{\partial(-\log p_\theta(i_y|x))}{\partial z_i} = -\frac{1}{p_\theta(i_y|x)}\frac{\partial p_\theta(i_y|x)}{\partial z_i} \tag{23}$$

where $p_\theta(i_y|x)$ is the output probability of the correct label's index i_y. Using the property of Softmax, we have

$$\frac{\partial p_\theta(i_y|x)}{\partial z_i} = \begin{cases} p_\theta(i_y|x)(1 - p_\theta(i_y|x)), & \text{when } i = i_y,\\ -p_\theta(i_y|x)p_\theta(i|x), & \text{when } i \neq i_y.\end{cases} \tag{24}$$

So the gradient become

$$\frac{\partial\mathcal{L}(\theta;x)}{\partial z_i} = (p_\theta(1|x), p_\theta(2|x), ..., p_\theta(i_y|x) - 1, p_\theta(i_{y+1}|x), ..., p_\theta(n|x))^\top. \tag{25}$$

So the gradient norm of $\frac{\partial \mathcal{L}}{\partial z}$ can be bounded as

$$\begin{aligned}\left\|\frac{\partial \mathcal{L}(\theta;x)}{\partial z}\right\|_2^2 &= (p_\theta(i_y|x)-1)^2 + \sum_{i\neq y} p_\theta(i|x)^2 \\ &\leq (1-p_\theta(i_y|x))^2 + \left(\sum_{i\neq y} p_\theta(i|x)\right)^2 \\ &= 2(1-p_\theta(i_y|x))^2,\end{aligned}$$

which finally has $\left\|\frac{\partial \mathcal{L}(\theta;x)}{\partial z}\right\|_2 \leq \sqrt{2}(1-p_\theta(i_y|x))$. Because $\mathrm{GMS}(x,y) = p_\theta(i_y|x) - \max_{i\neq i_y} p_\theta(i|x) \leq p_\theta(i_y|x) \leq 1$, so we have

$$\left\|\frac{\partial \mathcal{L}(\theta;x)}{\partial z}\right\|_2 \leq \sqrt{2}(1-p_\theta(i_y|x)) \leq \sqrt{2}(1-\mathrm{GMS}(x,y)).$$

To get the lower bound, we first define a mean probability as $\bar{p} = \frac{1}{n-1}\sum_{i\neq i_y} p_\theta(i|x) = \frac{1-p_\theta(i_y|x)}{n-1}$, and we have

$$\begin{aligned}\sum_{i\neq i_y} p_\theta(i|x)^2 &= \sum_{i\neq y}(p_\theta(i|x)-\bar{p}+\bar{p})^2 \\ &= \sum_{i\neq i_y}(p_\theta(i|x)-\bar{p})^2 + 2\bar{p}\sum_{i\neq i_y}(p_\theta(i|x)-\bar{p}) + \sum_{i\neq i_y}\bar{p}^2 \\ &= \sum_{i\neq i_y}(p_\theta(i|x)-\bar{p})^2 + (n-1)\bar{p}^2.\end{aligned}$$

So for the gradient norm of $\frac{\partial \mathcal{L}}{\partial z}$, we have

$$\begin{aligned}\left\|\frac{\partial \mathcal{L}(\theta;x)}{\partial z}\right\|_2 &= (1-p_\theta(i_y|x))^2 + \sum_{i\neq i_y}(p_\theta(i|x)-\bar{p})^2 + (n-1)\bar{p} \\ &= (1-p_\theta(i_y|x))^2 + \frac{(1-p_\theta(i_y|x))^2}{n-1} + \sum_{i\neq i_y}(p_\theta(i|x)-\bar{p})^2 \\ &\geq \frac{n}{n-1}(1-p_\theta(i_y|x))^2.\end{aligned}$$

From the definition of GMS and $\max_{i\neq i_y} p_\theta(i|x) \leq \sum_{i\neq i_y} p_\theta(i|x) = 1 - p_\theta(i_y|x)$, we have

$$\begin{aligned}&\mathrm{GMS}(x,y) = p_\theta(i_y|x) - \max_{i\neq i_y} p_\theta(i|x) \geq p_\theta(i_y|x) - (1-p_\theta(i_y|x)) \\ \Rightarrow &p_\theta(i_y|x) \leq \frac{1+\mathrm{GMS}(x,y)}{2} \\ \Rightarrow &1-p_\theta(i_y|x) \geq 1 - \frac{1+\mathrm{GMS}(x,y)}{2} = \frac{1-\mathrm{GMS}(x,y)}{2}.\end{aligned}$$

So the lower bound for the norm of $\frac{\partial \mathcal{L}}{\partial z}$ is

$$\left\|\frac{\partial \mathcal{L}(\theta;x)}{\partial z}\right\|_2 \geq \frac{\sqrt{n/(n-1)}}{2}(1-\mathrm{GMS}(x,y)).$$

So the lower bound and upper bound of the $\left\|\frac{\partial \mathcal{L}}{\partial z}\right\|_2$ is proved. Then, the gradient can be represented as

$$\begin{aligned}\|\nabla \mathcal{L}(\theta;x)\| &= \left\|\frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial \theta}\right\| \\ &\leq \lambda_{\max}\left\|\frac{\partial \mathcal{L}}{\partial z}\right\| \\ &\leq \lambda_{\max}\sqrt{2}(1-\mathrm{GMS}(x,y)),\end{aligned}$$

and for the lower bound, it satisfies

$$\begin{aligned}\|\nabla\mathcal{L}(\theta;x)\| &= \left\|\frac{\partial\mathcal{L}}{\partial z}\frac{\partial z}{\partial\theta}\right\| \\ &\geq \lambda_{\min}\left\|\frac{\partial\mathcal{L}}{\partial z}\right\| \\ &\geq \lambda_{\min}\frac{\sqrt{n/(n-1)}}{2}(1-\mathrm{GMS}(x,y)).\end{aligned}$$

The proof is completed. □

F Proof of Theorem B.2

Proof. First, we consider the case of a single query with perturbation u in the (2) (i.e., $q=1$), for which we denote the two-point central-difference as

$$D_\mu(u) := \frac{\mathcal{L}(\theta+\mu u;\mathcal{B})-\mathcal{L}(\theta-\mu u;\mathcal{B})}{2\mu}. \tag{26}$$

We consider a standard assumption in ZO analysis that the loss function $\mathcal{L}$ is twice continuously differentiable with bounded Hessian $\|\nabla^2\mathcal{L}(\theta;\mathcal{B})\|_{\mathrm{op}}\leq L$. Then, from the Taylor expansion, there exist $\zeta,\xi\in(0,1)$ such that

$$\mathcal{L}(\theta+\mu u;\mathcal{B}) = \mathcal{L}(\theta;\mathcal{B})+\mu\nabla\mathcal{L}(\theta;\mathcal{B})^\top u+\frac{\mu^2}{2}u^\top\nabla^2\mathcal{L}(\theta+\zeta\mu u;\mathcal{B})u, \tag{27}$$

and

$$\mathcal{L}(\theta-\mu u;\mathcal{B}) = \mathcal{L}(\theta;\mathcal{B})-\mu\nabla\mathcal{L}(\theta;\mathcal{B})^\top u+\frac{\mu^2}{2}u^\top\nabla^2\mathcal{L}(\theta-\xi\mu u;\mathcal{B})u. \tag{28}$$

Substituting (27) and (28) into (26), we have

$$D_\mu(u) = \nabla\mathcal{L}(\theta;\mathcal{B})^\top u+R_\mu(u), \tag{29}$$

with

$$R_\mu(u) = \frac{\mu}{4}u^\top\Big(\nabla^2\mathcal{L}(\theta+\zeta\mu u;\mathcal{B})-\nabla^2\mathcal{L}(\theta-\xi\mu u;\mathcal{B})\Big)u. \tag{30}$$

With the two-point central-difference (26), the gradient estimation for q queries can be written as

$$\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}) = \frac{1}{q}\sum_{i=1}^{q}\frac{\mathcal{L}(\theta+\mu u_i;\mathcal{B})-\mathcal{L}(\theta-\mu u_i;\mathcal{B})}{2\mu}u = \frac{1}{q}\sum_{i=1}^{q}D_\mu(u_i)u_i. \tag{31}$$

Then, using (29) we have

$$\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}) = \frac{1}{q}\sum_{i=1}^{q}\left((\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)u_i+R_\mu(u_i)u_i\right). \tag{32}$$

By forcing $\mu\to 0$, we further have

$$\lim_{\mu\to 0}\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B}) = \frac{1}{q}\sum_{i=1}^{q}(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)u_i.$$

The expected norm of the estimated gradient when $\mu\to 0$ can be computed as

$$\begin{aligned}\lim_{\mu\to 0}\mathbb{E}\left[\left\|\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right\|^2\right] &= \frac{1}{q^2}\mathbb{E}\left[\left\|\sum_{i=1}^{q}(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)u_i\right\|\right] \\ &= \frac{1}{q^2}\mathbb{E}\left[\sum_{i=1}^{q}\sum_{j=1}^{q}(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_j)u_i^\top u_j\right].\end{aligned} \tag{33}$$

For the RHS of (33), when $i = j$, we have

$$\begin{aligned}\mathbb{E}\left[(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)u_i^\top u_i\right] &= \mathbb{E}[(u_i^\top e)\|u_i\|^2]\,\|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2 \\ &= (d+2)\,\|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2,\end{aligned} \tag{34}$$

when $i \neq j$, we also have

$$\begin{aligned}\mathbb{E}\left[(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_j)u_i^\top u_j\right] &= \mathbb{E}_{u_i}\left[(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)\cdot\mathbb{E}_{u_j}\left[(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_j)u_i^\top u_j\right]\right] \\ &= \mathbb{E}_{u_i}\left[(\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)(u_i^\top\nabla\mathcal{L}(\theta;\mathcal{B}))\right] \\ &= \|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2\end{aligned} \tag{35}$$

Because there are q terms of (34) and $q(q-1)$ terms of (35), so we have

$$\begin{aligned}\lim_{\mu\to 0}\mathbb{E}\left[\left\|\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right\|^2\right] &= \frac{q(d+2)\,\|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2 + q(q-1)\,\|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2}{q^2} \\ &= \left(1+\frac{d+1}{q}\right)\|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2.\end{aligned} \tag{36}$$

For the variance term, we have

$$\begin{aligned}\lim_{\mu\to 0}\mathrm{Cov}_{\mathcal{B}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right) &= \frac{1}{q^2}\sum_{i=1}^{q}\mathrm{Cov}_{\mathcal{B}}\left((\nabla\mathcal{L}(\theta;\mathcal{B})^\top u_i)u_i\right) \\ &= \frac{1}{q}\mathrm{Cov}_{\mathcal{B}}\left((\nabla\mathcal{L}(\theta;\mathcal{B})^\top u)u\right) \\ &= \frac{1}{q}\left(\|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2 I + \nabla\mathcal{L}(\theta;\mathcal{B})\nabla\mathcal{L}(\theta;\mathcal{B})^\top\right)\end{aligned} \tag{37}$$

Because the variance is the trace of the covariance matrix, we have

$$\lim_{\mu\to 0}\mathrm{Var}_{\mathcal{B}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right) = \mathrm{Tr}\left(\lim_{\mu\to 0}\mathrm{Cov}_{\mathcal{B}}\left(\widehat{\nabla}\mathcal{L}(\theta;\mathcal{B})\right)\right) = \frac{d+1}{q}\,\|\nabla\mathcal{L}(\theta;\mathcal{B})\|^2.$$

The proof is completed.

□

G ZO Estimation MSE with Difficulty-Aware Reweighting

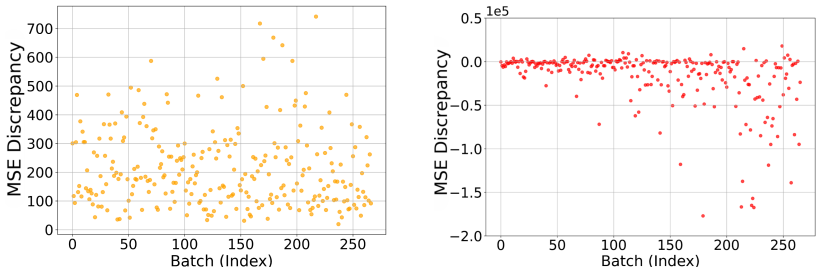
While Figure 3 compares the full-gradient referenced MSE of vanilla ZO (using $\mathcal{L}(\theta;\mathcal{B})$) and DaZO (using the reweighted loss $\mathcal{L}_w(\theta;\mathcal{B})$), here we further investigate the effect of the difficulty-aware reweighting. To this end, we compute the difference between the full-gradient referenced MSE of vanilla ZO and that of DaZO using the proposed difficulty-aware reweighting. Meanwhile, we also compute the difference between the full-gradient referenced MSE of vanilla ZO and that of the ZO method that uses a random reweighting, with the weighting coefficients following a uniform distribution in $[0, 1]$. The results on SST-5 with RoBERTa-Large are provided in Figure 11.

It can be seen that, while DaZO using the proposed difficulty-aware reweighting has consistently lower MSE (w.r.t. the global gradient $\nabla\mathcal{L}(\theta)$) than vanilla ZO, using a random reweighting can lead to higher MSE. These results verify the effectiveness of the proposed difficulty-aware reweighting in reducing the ZO gradient estimation error w.r.t. the global gradient $\nabla\mathcal{L}(\theta)$.

H Experiment Setup

H.1 Dataset

For RoBERTa-Large, we consider classification datasets including SST-2, SST-5, MNLI, SNLI, RTE and TREC. For OPT-1.3B/13B and Llama3-8B, we consider the datasets including SST-2, RTE, CB, BoolQ, WSC, WIC, MultiRC, Copa, and ReCoRD. Meanwhile, we also evaluate the proposed method on a generation task DROP with OPT-1.3B. Moreover, we consider the SST-2, RTE, WSC, and WIC tasks for OPT-30B.



(a) MSE discrepancy between MeZO and DaZO

(b) MSE discrepancy between MeZO and randomly weighted ZO

Figure 11: (a) Difference between the full-gradient referenced MSE of vanilla ZO and that of DaZO using the proposed difficulty-aware reweighting. (b) Difference between the full-gradient referenced MSE of vanilla ZO and that of the ZO method using a random reweighting. While DaZO using the proposed difficulty-aware reweighting has consistently lower MSE (w.r.t. the global gradient $\nabla\mathcal{L}(\theta)$) than vanilla ZO, using a random reweighting can lead to higher MSE.

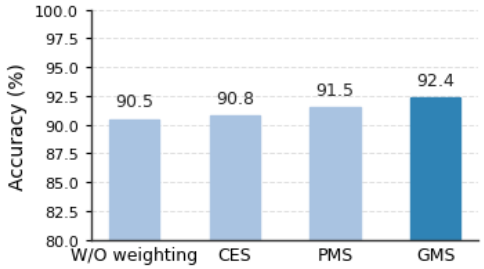


Figure 12: Performance of DaZO with three difficulty scores on the SST-2 task, using RoBERTa-Large (k=16). "w/o weighting" denotes the vanilla ZO method without using sample-wise reweighting.

H.2 Hyperparameter Setting

We use the hyperparameter settings in Tables 7 to 9 for experiments on RoBERTa-Large, OPT-1.3B/13B/30B and Llama3-8B. We use constant learning rate for all zeroth-order methods. For all experiments, we evaluate the model on validation sets every 1000 steps and save the best validation checkpoint, and all experiments use 20K steps.

I More Experiment Results

I.1 More Experimental Results on RoBERTa-Large

Figure 12 shows the performance of DaZO with three difficulty scores.

Figure 13 compares the full fine-tuning results of MeZO, HiZOO, LOZO, PaZO, and DaZO on RoBERTa-Large with $k = 16$. More detailed results are provided in Table 1.

Table 10 presents the results on RoBERTa-Large with $k = 512$, in comparison with FO-Adam, MeZO, HiZOO, LOZO. It can be seen that the proposed DaZO achieves the best performance among the compared ZO fine-tuning methods.

Table 7: Hyperparameter setting used for the RoBERTa-Large experiments. All experiments use a constant learning rate schedule.

Experiment	Hyperparameters	Values
MeZO	Batch Size	64
	Learning Rate	{1e-7, 1e-6, 1e-5}
	μ	1e-3
	Weight Decay	0
MeZO(prefix)	Batch Size	64
	Learning Rate	{1e-3, 5e-3, 1e-2}
	μ	1e-1
	Weight Decay	0
	# prefix tokens	5
MeZO(LoRA)	Batch Size	64
	Learning Rate	{1e-5, 5e-5, 1e-4}
	μ	1e-3
	Weight Decay	0.1
	(r, α)	(8, 16)
PaZO	Batch Size	64
	Learning Rate	{1e-7, 1e-6, 1e-5}
	μ	1e-3
	Weight Decay	0
PaZO(prefix)	Batch Size	64
	Learning Rate	{1e-3, 5e-3, 1e-2}
	μ	1e-1
	Weight Decay	0
	# prefix tokens	5
PaZO(LoRA)	Batch Size	64
	Learning Rate	{1e-5, 5e-5, 1e-4}
	μ	1e-3
	Weight Decay	0.1
	(r, α)	(8, 16)
DaZO	Batch Size	64
	Learning Rate	{1e-5, 2e-5, 5e-5}
	μ	1e-3
	Weight Decay	0.15
	λ	0.01
DaZO(prefix)	Batch Size	64
	Learning Rate	{1e-3, 5e-3, 1e-2}
	μ	1e-1
	Weight Decay	0
	# prefix tokens	5
	λ	0.01
DaZO(LoRA)	Batch Size	64
	Learning Rate	{1e-5, 2e-5, 5e-5}
	μ	1e-3
	Weight Decay	0.1
	(r, α)	(8, 16)
	λ	0.01
FT	Batch Size	8
	Learning Rate	{1e-5, 3e-5, 5e-5}
	Weight Decay	0

Table 8: Hyperparameter setting used for OPT-1.3B experiments. All experiments use a constant learning rate schedule.

Experiment	Hyperparameters	Values
MeZO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	μ	1e-3
MeZO(prefix)	Batch Size	16
	Learning Rate	{5e-3, 1e-2, 5e-2}
	μ	1e-1
	# prefix tokens	5
MeZO(LoRA)	Batch Size	16
	Learning Rate	{1e-5, 5e-5, 1e-4}
	μ	1e-2
	(r, α)	(8, 16)
PaZO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	μ	1e-3
PaZO(prefix)	Batch Size	16
	Learning Rate	{5e-3, 1e-2, 5e-2}
	μ	1e-1
	# prefix tokens	5
PaZO(LoRA)	Batch Size	16
	Learning Rate	{1e-5, 5e-5, 1e-4}
	μ	1e-2
	(r, α)	(8, 16)
DaZO	Batch Size	16
	Learning Rate	{1e-5, 2e-5, 5e-5}
	μ	1e-3
	λ	0.01
DaZO(prefix)	Batch Size	16
	Learning Rate	{5e-3, 1e-2, 5e-2}
	μ	1e-2
	# prefix tokens	5
	λ	0.01
DaZO(LoRA)	Batch Size	16
	Learning Rate	{1e-5, 2e-5, 5e-5}
	μ	1e-3
	(r, α)	(8, 16)
	λ	0.01

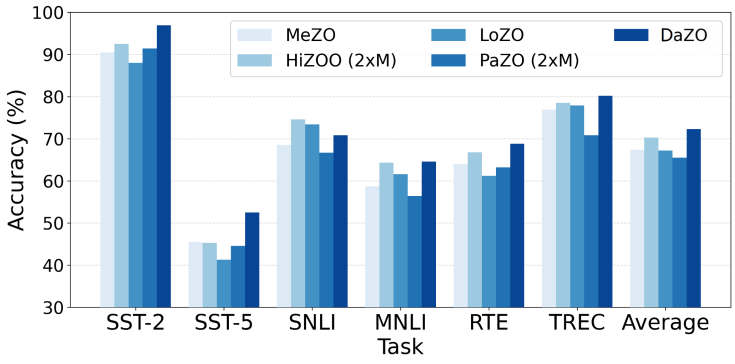


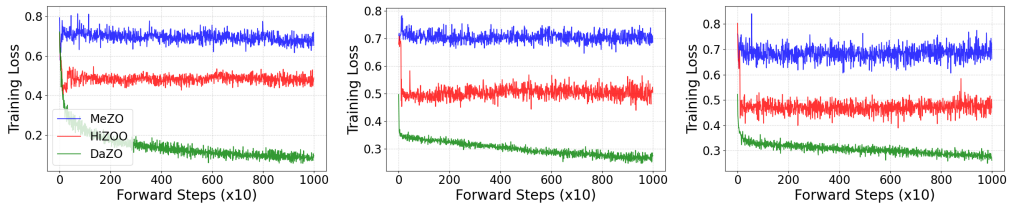
Figure 13: Accuracy comparison on RoBERTa-Large (350M, k=16) across multiple tasks.

Table 9: Hyperparameter setting used for Llama3-8B and OPT-13B experiments. All experiments use a constant learning rate schedule.

Experiment	Hyperparameters	Values
MeZO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	μ	1e-3
LOZO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	μ	1e-3
	Rank(r)	{1, 2, 4}
	Interval(v)	{50, 100}
DaZO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	μ	1e-3
	λ	0.05

Table 10: Results on RoBERTa-Large (350M) on six downstream tasks ($k = 512$).Results of DaZO are averaged over five runs.

Task	SST-2	SST-5	SNLI	MNLI	RTE	TREC	Average
Zero-shot	79.0	35.5	50.2	48.8	51.4	32.0	49.5
LP	91.3	51.7	80.9	71.5	73.1	89.4	76.3
FO-Adam	94.5	55.9	88.7	84.4	82.7	97.3	83.9
MeZO	93.3	53.2	83.0	78.3	78.6	94.3	80.1
HiZOO	95.5	53.2	82.6	77.7	80.0	94.6	80.6
LOZO	94.1	53.0	85.4	80.4	79.7	95.5	81.4
DaZO	94.3	55.3	83.9	80.6	80.5	94.8	81.6



(a) OPT-1.3B on SST-2 (b) OPT-1.3B on RTE (c) OPT-1.3B on BoolQ

Figure 14: Loss curves during training for different ZO methods on the SST-2, RTE and BoolQ tasks with OPT-1.3B.

I.2 Loss Curve Comparison

Figure 14 compares the loss curves during training of MeZO, HiZOO and DaZO on the SST-2, RTE and BoolQ tasks with OPT-1.3B. It can be seen that DaZO shows significantly faster convergence than MeZO and HiZOO.

I.3 Ablation Study

We further examine DaZO's sensitivity to the hyperparameter λ. DaZO is stable across a broad range and performs best for $\lambda \in [0.01, 0.03]$. Even outside this interval, DaZO consistently outperforms the other ZO baselines, which indicates low sensitivity to λ. Figure 15 shows the accuracy on SST-2 when RoBERTa-Large with different λ values.

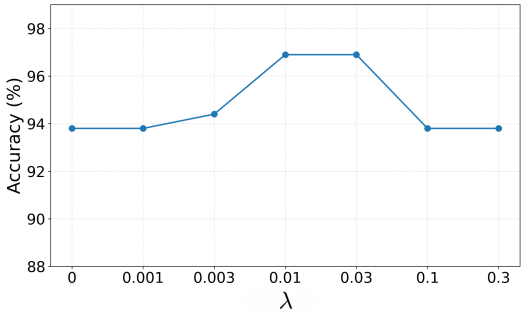


Figure 15: The effect of the parameter λ on the performance of DaZO on SST-2 with RoBERTa-Large.

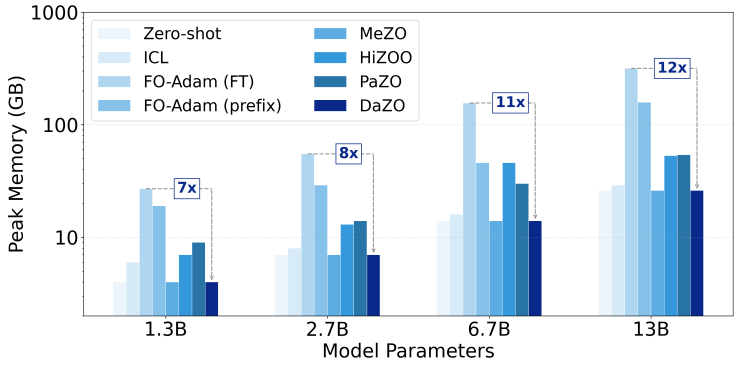


Figure 16: GPU peak memory usage of different methods with different OPT models on MultiRC (400 tokens per example on average). See Table 11 for details.

Table 11: Peak memory on the MultiRC (average tokens=400) task with different OPT models on NVIDIA A100.

Method	OPT-1.3B	OPT-2.7B	OPT-6.7B	OPT-13B
Zero-shot/MeZO/DaZO	4GB	7GB	14GB	26GB
ICL	6GB	8GB	16GB	29GB
HiZOO	7GB	13GB	46GB	53GB
PaZO	9GB	14GB	30GB	54GB
FO-Adam (FT)	27GB	55GB	156GB	316GB
FO-Adam (prefix)	19GB	29GB	46GB	158GB

I.4 Memory Usage Comparison

As shown in Figure 16, DaZO and MeZO incur inference-level peak memory that matches that of Zero-shot, substantially lower than other methods. Table 11 compares the peak memory overhead of Zero-shot, ICL, MeZO, HiZOO, PaZOO, DaZO, FO-Adam (FT) and FO-Adam (prefix). All measurements were conducted with a per-device batch-size of 1 without employing memory optimization strategies such as gradient checkpointing. The memory is monitored via Nvidia's nvidia-smi command. It can be seen that both MeZO and DaZO have the same memory overhead as the Zero-shot method, which is substantially lower than that of the first-order methods. Moreover, as HiZOO and PaZOO require storing diagonal Hessian information, which incurs a memory overhead of 2 times the model weight memory M.

I.5 Wall-clock time Comparison

Table 12 shows the wall-clock time per step for MeZO, HiZOO, and DaZO on SST-2 across different model scales.

Table 12: Comparison of average wall-clock time per step of MeZO, HiZOO and DaZO on SST-2. The batch-size for the relatively smaller RoBERTa-Large model is 64, while that for the other larger models is 16.

Model	MeZO	HiZOO	DaZO
RoBERTa-Large	0.21s	0.30s	0.21s
OPT-1.3B	0.26s	0.38s	0.26s
Llama3-8B	0.75s	1.11s	0.75s
OPT-13B	1.11s	1.52s	1.11s