MSZO: SCHEDULED MULTI-SCALE ZEROTH-ORDER FINE-TUNING OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Compared with first-order (FO) backpropagation, zeroth-order (ZO) optimization is much more memory-efficient in fine-tuning large language models (LLMs) by estimating gradients using only forward passes. Existing ZO methods typically use a fixed perturbation scale for finite-difference gradient estimation throughout training, which implicitly assumes a stationary bias-variance trade-off across optimization stages. In this work, we show that this trade-off changes substantially during LLM fine-tuning. By probing checkpoints along LLM fine-tuning trajectories, we find a pronounced stage-dependent perturbation-scale effect: a small perturbation scale yields lower variance and lower FO-gradient referenced mean-squared error (MSE) early in training, but both can exceed those of a larger scale by the middle stage. In contrast, the larger scale provides more stable estimates at later stages at the cost of higher bias. Motivated by this finding, we propose MSZO, a scheduled multi-scale ZO optimizer that uses a small perturbation scale in the early stage and subsequently combines small- and large-scale estimates along the same random direction. This design dynamically balances bias and variance over the course of training. Experiments on RoBERTa-Large, OPT-1.3B, Llama3-8B, and Qwen3-4B demonstrate consistent improvement of MSZO over single-scale ZO fine-tuning. Unlike the common practice of using a fixed perturbation radius throughout fine-tuning, our results reveal that the appropriate scale in ZO optimization is inherently stage-dependent. Code is available at `https://anonymous.4open.science/r/MSZO-4FE9/`.

1 INTRODUCTION

Fine-tuning large language models (LLMs) with first-order (FO) optimizers requires backpropagation through the entire model, which results in substantial memory overhead from storing intermediate activations and optimizer states. Zeroth-order (ZO) optimization provides an appealing alternative, where model updates are estimated solely from forward evaluations of perturbed model parameters. This reduces training memory toward the inference-level footprint, which can be over $10\times$ lower than that of FO fine-tuning (Malladi et al., 2023; Zhang et al., 2024b). Therefore, ZO optimization is particularly promising for LLM fine-tuning on resource-constrained devices.

Recently, a number of methods have been developed to improve ZO fine-tuning of LLMs through low-dimensional or low-rank perturbations (Guo et al., 2025; Chen et al., 2025; Yu et al., 2025; Liu et al., 2024; Lin et al., 2026), curvature-aware ZO optimization (Zhao et al., 2025b;a), and variance reduction (Gautam et al., 2024). Despite these progresses, the perturbation scale, which is a fundamental component of ZO gradient estimation, has received little attention. Most existing ZO fine-tuning methods keep the perturbation scale fixed throughout training, without explicitly accounting for changes in the local loss landscape.

In this work, we show that this does not hold in LLM fine-tuning. Specifically, we analyze the effect of perturbation scale on ZO gradient estimates in LLM fine-tuning on model checkpoints along standard FO fine-tuning trajectories. Our analysis reveals a pronounced *stage-dependent perturbation-scale effect*: At early stages, a small perturbation radius consistently produces lower estimation variance and lower FO-gradient referenced mean-squared error (MSE) than a larger radius. However, as fine-tuning progresses, the behavior changes markedly: the variance and MSE of the small-scale estimator can exceed those of a large-scale estimator, whereas the large-scale estimator becomes

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.

more stable. Furthermore, the larger perturbation has larger bias, which indicates that neither scale uniformly dominates throughout training. The results suggest that the appropriate perturbation scale depends on the optimization stage.

Based on these findings, we propose a scheduled multi-scale ZO optimizer (MSZO). MSZO begins with a small perturbation radius to exploit its low bias and variance in the early stage. After a prescribed transition point, it evaluates both a small radius and a larger radius and combines their directional estimates to dynamically balance bias and variance of ZO gradient estimation over the later stage.

The main contributions of this work are as follows:

- We identify a *stage-dependent perturbation-scale effect* in ZO fine-tuning of LLMs. Small-scale perturbations provide lower variance estimates early in training than larger-scale perturbations, yet larger perturbations offer more stable estimates at later stages with larger bias.
- Based on this finding, we develop MSZO, a simple scheduled multi-scale ZO method that varies the perturbation strategy over the course of optimization. It starts with a small-scale and later integrates estimates from both small and large scales, which allows the optimizer to exploit their complementary bias-variance properties at different training stages.
- Across RoBERTa-Large, OPT-1.3B, Llama3-8B, and Qwen3-4B, MSZO delivers consistent gains over strong single-scale ZO methods on diverse downstream tasks, which demonstrates the effectiveness of the proposed scheduled multi-scale method across different model architectures and sizes.

Our results suggest a broader principle for ZO optimization based network learning: the perturbation scale should adapt to the evolving geometry of the optimization landscape rather than remain fixed throughout training.

2 Related Work

Fine-tuning LLMs with FO optimizers such as Adam (Kingma & Ba, 2014) requires substantial memory for backpropagation. Parameter-efficient fine-tuning (PEFT) methods (Houlsby et al., 2019; Hu et al., 2022; Li & Liang, 2021) largely reduce the trainable parameters but still rely on FO backpropagation with substantial memory overhead required for storing intermediate activations.

ZO fine-tuning of LLMs. ZO optimization provides a memory-efficient alternative for LLM fine-tuning using only forward evaluations. Classical ZO methods have been extensively studied in stochastic and derivative-free optimization (Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017; Chen et al., 2019; Liu et al., 2019a; 2020; Duchi et al., 2015) and widely applied in modern machine learning (Zhang et al., 2022b; Wang et al., 2022; Shu et al., 2023; Zhang et al., 2024a; 2026; Qin et al., 2026). Their application to LLMs was popularized by MeZO (Malladi et al., 2023), which enables full LLM fine-tuning with an inference-level memory footprint. Despite this advantage, the high variance of ZO gradient estimates remains a major limitation, particularly in the high-dimensional parameter space of modern LLMs.

A growing body of work therefore exploits the intrinsic structure of LLM adaptation to improve ZO fine-tuning. For example, LOZO (Chen et al., 2025), SubZero (Yu et al., 2025) and TeZO (Sun et al., 2025) exploit low-rank structures to reduce the dimension of gradient estimation. Moreover, the works (Guo et al., 2025) and (Liu et al., 2024) considers sparse ZO updates to selected subsets of parameters. More recent method AGZO (Lin et al., 2026) incorporates forward-pass information to construct informative perturbation spaces, which restricts perturbations to activation-informed subspaces.

Another line of work improves ZO fine-tuning through curvature information, multi-queries, or more structured estimators. HiZOO (Zhao et al., 2025b) estimates diagonal Hessian to account for heterogeneous curvature across parameter dimensions, while (Zhao et al., 2025a) further analyzes preconditioned ZO optimization. FZOO (Dang et al., 2025) uses multiple queries to probe the loss curvature for adaptive learning-rate reweighting. (Zhang et al., 2025) considers a block-wise Hessian structure and learning efficient perturbation strategies. MUZO (Peng et al., 2025) reduces estimation variance by multiple queries and momentum-based optimization. PseuZO (Yue et al., 2025) modifies

the estimation target by exploiting the compositional structure of neural networks and averaging intermediate Jacobian estimates. Recent approach ZO-Muon (Lang et al., 2026) further combines low-dimensional subspace estimation with spectral processing of noisy ZO updates.

While these methods focus on which spaces or which directions to perturb, or on how to precondition the estimates, our work studies at what perturbation scale the objective should be probed, and whether this scale should remain unchanged throughout fine-tuning.

Perturbation scale in ZO optimization. The perturbation radius controls a fundamental bias-variance trade-off in finite-difference ZO estimation. Smaller perturbations provide more local estimates with lower smoothing bias, whereas larger perturbations can suppress fine-scale fluctuation and estimation noise (Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017). Classical stochastic approximation methods such as SPSA typically employ a diminishing perturbation sequence to improve local approximation as optimization proceeds (Spall, 1998). In contrast, modern ZO based network training and LLM fine-tuning generally treat the perturbation radius as a fixed hyperparameter (Chen et al., 2024; Malladi et al., 2023). Our work identifies a clear stage-dependent scale preference in ZO LLM fine-tuning: small perturbations are more reliable in the early stage of fine-tuning but larger perturbations are more reliable in later stages. Our method MSZO therefore transitions from small-scale estimation to a mixture of small- and large-scale estimates, which adapts the bias-variance trade-off to the evolving optimization landscape rather than monotonically shrinking the perturbation radius.

3 Analysis on the Perturbation-Scale Effect in ZO Based LLM Fine-tuning

3.1 Preliminary of Randomized Gradient Estimation

Basically, ZO optimization estimates descent directions solely from forward function evaluations. A widely adopted ZO gradient estimator is the randomized gradient estimator (RGE) based on symmetric two-point finite differences (SPALL, 1992; Duchi et al., 2015; Nesterov & Spokoiny, 2017). Specifically, let $\mathcal{L}(\boldsymbol{\theta};\mathcal{B})$ denote the training objective on a mini-batch $\mathcal{B}$ with model parameters $\boldsymbol{\theta} \in \mathbb{R}^d$. RGE samples a random perturbation direction $\boldsymbol{z} \in \mathbb{R}^d$, e.g., $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and evaluates the objective at two symmetrically perturbed points, $\mathcal{L}(\boldsymbol{\theta}+\epsilon\boldsymbol{z};\mathcal{B}))$ and $\mathcal{L}(\boldsymbol{\theta}-\epsilon\boldsymbol{z};\mathcal{B}))$, where $\epsilon > 0$ denotes the perturbation scale. The corresponding directional derivative is estimated by

$$d_\epsilon = \frac{\mathcal{L}(\boldsymbol{\theta}+\epsilon\boldsymbol{z};\mathcal{B}) - \mathcal{L}(\boldsymbol{\theta}-\epsilon\boldsymbol{z};\mathcal{B})}{2\epsilon}, \tag{1}$$

and the randomized gradient estimate is given by

$$\widehat{\boldsymbol{g}}_\epsilon = d_\epsilon \boldsymbol{z}. \tag{2}$$

The symmetric construction in (1) cancels the first-order truncation error, yielding an $O(\epsilon^2)$ approximation to the directional derivative. Under standard smoothness conditions, a Taylor expansion gives $d_\epsilon = \boldsymbol{z}^\top \nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})) + O(\epsilon^2)$, up to factors depending on the perturbation direction and higher-order derivatives. Hence, for sufficiently small ϵ, RGE provides a stochastic approximation to the FO gradient using only two forward evaluations.

More precisely, define the Gaussian-smoothed objective

$$\mathcal{L}_\epsilon(\boldsymbol{\theta};\mathcal{B}) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\mathcal{L}(\boldsymbol{\theta}+\epsilon\boldsymbol{z};\mathcal{B})\right]. \tag{3}$$

For Gaussian perturbations, the two-point estimator (2) satisfies $\mathbb{E}_{\boldsymbol{z}}\left[\widehat{\boldsymbol{g}}_\epsilon\right] = \nabla\mathcal{L}_\epsilon(\boldsymbol{\theta};\mathcal{B})$. Thus RGE is an unbiased estimator of the gradient of a smoothed objective $\mathcal{L}_\epsilon$ rather than the gradient of $\mathcal{L}$. As $\epsilon \to 0$, $\mathcal{L}_\epsilon$ approaches the original objective under certain regularity conditions.

This formulation highlights the role of the perturbation scale ϵ. A smaller ϵ provides a more local approximation and reduces smoothing bias, whereas a larger ϵ probes the objective over a broader neighborhood and may suppress fine-scale variation in loss function. Therefore, the choice of ϵ directly determines the statistical and approximation properties of ZO gradient estimation. In the following, we show that these properties vary substantially over the course of LLM fine-tuning, which challenges the conventional practice of using a fixed perturbation scale throughout fine-tuning.



(a) SST-2 (b) RTE (c) WIC

Figure 1: ZO gradient estimation statistics across three tasks using OPT-1.3B, evaluated along FO SGD fine-tuning trajectories. Each column presents the results on one task. Rows show variance $V_{\epsilon,t}$, FO-gradient referenced MSE $E_{\epsilon,t}$ and squared bias $B_{\epsilon,t}$, respectively.

3.2 The Stage-Dependent Perturbation-Scale Effect of ZO Fine-tuning

First, we investigate how the statistical properties of ZO gradient estimation evolve over the course of LLM fine-tuning. To isolate the effect of the local optimization landscape from the dynamics induced by a specific ZO optimizer, we probe model checkpoints collected along a FO SGD fine-tuning trajectory. At each checkpoint $\boldsymbol{\theta}_t$ in the SGD trajectory, we compute the corresponding FO gradient $\boldsymbol{g}_t = \nabla \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}_t)$ and sample K Gaussian directions to construct ZO estimates $\{\widehat{\boldsymbol{g}}_{\epsilon,t}^{(k)}\}_{k=1}^K$ using (2). Let $\bar{\boldsymbol{g}}_{\epsilon,t} = \frac{1}{K}\sum_{k=1}^K \widehat{\boldsymbol{g}}_{\epsilon,t}^{(k)}$ denote the mean estimation at checkpoint $\boldsymbol{\theta}_t$ over K directions. We consider three complementary statistics on the ZO estimates $\{\widehat{\boldsymbol{g}}_{\epsilon,t}^{(k)}\}_{k=1}^K$, including variance

$$V_{\epsilon,t} = \frac{1}{K}\sum_{k=1}^{K} \left\|\widehat{\boldsymbol{g}}_{\epsilon,t}^{(k)} - \bar{\boldsymbol{g}}_{\epsilon,t}\right\|_2^2, \tag{4}$$

and FO-gradient referenced MSE and FO-gradient referenced squared bias

$$E_{\epsilon,t} = \frac{1}{K}\sum_{k=1}^{K} \left\|\widehat{\boldsymbol{g}}_{\epsilon,t}^{(k)} - \boldsymbol{g}_t\right\|_2^2, \qquad B_{\epsilon,t} = \left\|\bar{\boldsymbol{g}}_{\epsilon,t} - \boldsymbol{g}_t\right\|_2^2. \tag{5}$$

The variance $V_{\epsilon,t}$ measures the variability induced by random perturbations, whereas $E_{\epsilon,t}$ measures the discrepancy from the FO gradient. The quantity $B_{\epsilon,t}$ measures the empirical squared bias of the ZO estimator. Figure 1 presents these quantities on three tasks using the OPT-1.3B model. We compare two fixed-scale estimators with $\epsilon_s = 10^{-3}$ and $\epsilon_l = 5 \times 10^{-3}$, respectively, and a mixed-scale estimator $0.8\widehat{\boldsymbol{g}}_{\epsilon_s,t} + 0.2\widehat{\boldsymbol{g}}_{\epsilon_l,t}$, where both gradient estimates are computed using the same Gaussian direction.

As shown in Figure 1, the small perturbation scale yields lower variance and lower FO-gradient referenced MSE at early stage checkpoints with smaller bias. However, in middle and later stages,

the variance of the small-scale estimator can exceed that of the large-scale estimator. The same trend appears in the FO-gradient referenced MSE. In contrast, the large-scale estimator becomes more stable at later stages. Although the large-scale estimator is more stable later in training, it generally has a larger bias with respect to the FO gradient. This reflects stronger smoothing over local loss landscape. Thus, the large scale does not simply dominate the small scale.

These results reveal a stage-dependent perturbation-scale effect that the preferred perturbation scale changes during the course of LLM fine-tuning. Small perturbations are more accurate and stable early in training, but become less reliable in later stages compared with larger perturbations. Large perturbations provide better stability at later stages, at the cost of higher smoothing bias. This motivates adapting perturbation scales at different stages of LLM fine-tuning.

4 The Proposed Scheduled Multi-Scale ZO Method

Motivated by the above finding, we propose MSZO, a scheduled multi-scale ZO method that adapts the perturbation scale over the course of fine-tuning. Specifically, we divide the fine-tuning into two stages, an early single-scale stage and a subsequent multi-scale stage. We consider two perturbation scales, $0 < \epsilon_s < \epsilon_l$, where ϵ_s and ϵ_l denote the small and large perturbation scales, respectively. Let T denote the total number of training iterations. For a sampled Gaussian perturbation direction $\boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, the ZO gradient estimation at iteration t is computed as

$$\widehat{\boldsymbol{g}}_t = \left[(1-\beta_t)d_{\epsilon_s} + \beta_t d_{\epsilon_l}\right]\boldsymbol{z}_t, \tag{6}$$

with

$$\beta_t = \begin{cases} 0, & t < T_1 \\ \frac{\beta_0}{T-T_1}(T-t), & T_1 \le t \le T, \end{cases} \tag{7}$$

where $\beta_0 \in [0,1]$ is the initial mixing coefficient between the small- and large-scale estimates in the second stage, and $0 < T_1 < T$ is the transition iteration between the two stages. The coefficient β_t decreases linearly from β_0 at T_1 to zero at the end of training. This schedule uses the large-scale estimate to improve stability in the second stage, while gradually recovering the more local and less biased small-scale estimate as optimization approaches convergence. The two directional derivative estimates d_{ϵ_s} and d_{ϵ_l} for the two perturbation scales are given by

$$d_{\epsilon_s} = \frac{\mathcal{L}(\boldsymbol{\theta}_t + \epsilon_s \boldsymbol{z}_t; \mathcal{B}_t) - \mathcal{L}(\boldsymbol{\theta}_t - \epsilon_s \boldsymbol{z}_t; \mathcal{B}_t)}{2\epsilon_s}, \quad d_{\epsilon_l} = \frac{\mathcal{L}(\boldsymbol{\theta}_t + \epsilon_l \boldsymbol{z}_t; \mathcal{B}_t) - \mathcal{L}(\boldsymbol{\theta}_t - \epsilon_l \boldsymbol{z}_t; \mathcal{B}_t)}{2\epsilon_l}. \tag{8}$$

During the first stage ($t < T_1$), MSZO reduces to the conventional two-point estimator with the small perturbation radius ϵ_s. In the second stage, it combines the small- and large-scale estimates, with β_t controlling their relative contributions. Note that, in the second stage ($T_1 \le t \le T$), we use a combination of the small and large scales rather than a single large scale. The reason is that, although the larger perturbation is more stable and yields lower variance, it incurs higher smoothing bias, as shown in Figure 1. In comparison, the small scale provides a more local estimate with lower smoothing bias. Combining the two therefore provides a more favorable bias-variance trade-off.

The proposed schedule (6) differs from conventional perturbation-decay strategies in ZO optimization, which typically decrease the perturbation radius to obtain increasingly local estimates (Spall, 1998). In contrast, our analysis shows that smaller perturbations are not uniformly preferable during LLM fine-tuning, as their estimation variance can increase substantially at later stages. MSZO combines small- and large-scale estimates to adapt the effective smoothing level to the evolving optimization landscape over training.

The proposed multi-scale estimator for the second stage can be viewed as an interpolation between two gradient estimators operating at different spatial resolutions. Denote $\widehat{\boldsymbol{g}}_t^s = d_{\epsilon_s}\boldsymbol{z}_t$ and $\widehat{\boldsymbol{g}}_t^l = d_{\epsilon_l}\boldsymbol{z}_t$, the multi-scale gradient estimation can be expressed as

$$\widehat{\boldsymbol{g}}_t = (1-\beta_t)\widehat{\boldsymbol{g}}_t^s + \beta_t \widehat{\boldsymbol{g}}_t^l, \tag{9}$$

for which the bias with respect to the FO gradient can be written as

$$\mathrm{Bias}(\widehat{\boldsymbol{g}}_t) = (1-\beta_t)\,\mathrm{Bias}(\widehat{\boldsymbol{g}}_t^s) + \beta_t\,\mathrm{Bias}(\widehat{\boldsymbol{g}}_t^l), \tag{10}$$

whilst the variance satisfies

$$\mathrm{Var}(\widehat{\boldsymbol{g}}_t) = (1-\beta_t)^2\,\mathrm{Var}(\widehat{\boldsymbol{g}}_t^s) + \beta_t^2\,\mathrm{Var}(\widehat{\boldsymbol{g}}_t^l) + 2\beta_t(1-\beta_t)\,\mathrm{Cov}(\widehat{\boldsymbol{g}}_t^s, \widehat{\boldsymbol{g}}_t^l). \tag{11}$$

Therefore, MSZO does not assume that either perturbation scale universally dominates the other. Instead, it exploits their complementary statistical properties. Early in training, the low variance and low bias of the small-scale estimator make d_{ϵ_s} preferable. As optimization proceeds and the variance of d_{ϵ_s} increases, incorporating d_{ϵ_l} can reduce sensitivity, while retaining d_{ϵ_s} prevents excessive smoothing bias. The scheduled combination thus adapts the effective bias-variance trade-off to the evolving loss landscape.

5 Convergence Analysis

We first characterize the bias of MSZO under standard assumptions and then establish its convergence guarantees.

Assumption 1 (L-Smoothness)**.** The loss $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is differentiable and L-smooth, $\|\nabla\mathcal{L}(\boldsymbol{x}) - \nabla\mathcal{L}(\boldsymbol{y})\| \le L\|\boldsymbol{x}-\boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$.

Assumption 2 (Bounded Stochastic Gradient Variance)**.** For every query point $\boldsymbol{\theta} \in \mathbb{R}^d$, $\mathbb{E}_{\mathcal{B}}\left[\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right] = \nabla\mathcal{L}(\boldsymbol{\theta})$, and $\mathbb{E}_{\mathcal{B}}\left[\|\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B}) - \nabla\mathcal{L}(\boldsymbol{\theta})\|^2\right] \le \sigma_B^2$, where $\nabla\mathcal{L}(\boldsymbol{\theta})$ is the global gradient and σ_B^2 is a constant.

Lemma 1 (Bias and Second Moment of MSZO)**.** *Suppose Assumptions 1 and 2 hold. For every $\boldsymbol{\theta} \in \mathbb{R}^d$, the bias of MSZO satisfies*

$$\|\mathbb{E}_{\mathcal{B},\boldsymbol{z}}[\widehat{\boldsymbol{g}}_t] - \nabla\mathcal{L}(\boldsymbol{\theta})\| \le L\sqrt{d}\bar{\epsilon}_\beta, \tag{12}$$

and the Second Moment of MSZO satisfies

$$\mathbb{E}_{\mathcal{B},\boldsymbol{z}}\|\widehat{\boldsymbol{g}}_t\|^2 \le 2(d+2)\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 + 4L^2d(d+2)\bar{\epsilon}_\beta^2 + d(d+2)\sigma_B^2, \tag{13}$$

where $\bar{\epsilon}_\beta = (1-\beta_0)\epsilon_s + \beta_0\epsilon_l$.

Remark 1. *Lemma 1 reveals the bias effect of the multi-scale schedule. With $\epsilon_t = (1-\beta_t)\epsilon_s + \beta_t\epsilon_l$,*

$$\|\mathbb{E}[\widehat{\boldsymbol{g}}_t] - \nabla\mathcal{L}(\boldsymbol{\theta}_t)\| \le L\sqrt{d}\,\epsilon_t.$$

Thus, $\beta_t = 0$ recovers the MeZO estimator with the smaller perturbation scale ϵ_s. During the second stage, decreasing β_t reduces the contribution of the large-scale estimator and tightens the bias upper bound toward $L\sqrt{d}\,\epsilon_s$.

Theorem 1 (Convergence of MSZO)**.** *Suppose Assumptions 1 and 2 hold, and the objective is lower bounded by $\mathcal{L}(\boldsymbol{\theta}^\star)$. Under update rule $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\widehat{\boldsymbol{g}}_t$, for any $0 < \eta < \frac{1}{2L(d+2)}$, the iterates generated by MSZO satisfy*

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 \le{}& \frac{\Delta_0}{\eta T\left[\frac{1}{2} - L\eta(d+2)\right]} + \frac{L^2d\left[\frac{1}{2} + 2L\eta(d+2)\right]}{\frac{1}{2} - L\eta(d+2)}\bar{\epsilon}_\beta^2 \\ &+ \frac{L\eta d(d+2)}{2\left[\frac{1}{2} - L\eta(d+2)\right]}\sigma_B^2,\end{aligned} \tag{14}$$

where $\Delta_0 = \mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}(\boldsymbol{\theta}^\star)$. In particular, choosing $\eta^\star = \frac{1}{2L(d+2)\left[1+\sqrt{1+\frac{dT\left(3L^2\bar{\epsilon}_\beta^2 + \sigma_B^2/2\right)}{4L(d+2)\Delta_0}}\right]}$ minimizes the above upper bound and yields

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 \le{}& 2\sqrt{\frac{4L(d+2)\Delta_0}{T}\left[\frac{4L(d+2)\Delta_0}{T} + 3L^2d\bar{\epsilon}_\beta^2 + \frac{d\sigma_B^2}{2}\right]} \\ &+ \frac{8L(d+2)\Delta_0}{T} + L^2d\bar{\epsilon}_\beta^2.\end{aligned} \tag{15}$$

Table 1: Experiments on RoBERTa-large (350M, $k = 16$). All reported numbers are averaged accuracy with standard deviation over 5 runs.

Task Type	SST-2	SST-5	SNLI	MNLI	RTE	TREC	Average
	— sentiment —		— natural language inference —			— topic —	
Zero-shot	79.0	35.5	50.2	48.8	51.4	32.0	49.5
LP	76.0 ($\pm$2.8)	40.3 ($\pm$1.9)	66.0 ($\pm$2.7)	56.5 ($\pm$2.5)	59.4 ($\pm$5.3)	51.3 ($\pm$5.5)	58.3
MeZO	90.5 ($\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$2.7)	67.4
HiZOO	92.5 ($\pm$0.9)	45.3 ($\pm$1.2)	**74.6** ($\pm$1.3)	64.3 ($\pm$2.0)	**66.8** ($\pm$1.2)	78.5 ($\pm$1.1)	70.3
LOZO	88.0 ($\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$3.2)	67.2
MSZO	**93.3** ($\pm$1.0)	**47.5** ($\pm$1.1)	72.4 ($\pm$1.9)	**68.9** ($\pm$2.4)	65.4 ($\pm$2.1)	**78.9** ($\pm$1.8)	**71.1**

Corollary 1. *Under the conditions of Theorem 1, let $\eta = \eta^\star$ be given in Theorem 1. For fixed perturbation scales and bounded stochastic gradient variance, we have*

$$\eta^\star = O(T^{-1/2}), \tag{16}$$

and

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 \le L^2 d\bar{\epsilon}_\beta^2 + O(T^{-1/2}). \tag{17}$$

Therefore, MSZO achieves an $O(T^{-1/2})$ convergence rate in the average expected squared gradient norm, up to a perturbation-induced error floor of order $O(L^2 d\bar{\epsilon}_\beta^2)$.

The detailed proof is provided in Appendix A.

6 Experiments

We evaluate MSZO on RoBERTa-Large (Liu et al., 2019b), OPT-1.3B (Zhang et al., 2022a), Qwen3-4B (Yang et al., 2025), and Llama3-8B (Grattafiori et al., 2024). We compare against MeZO (Malladi et al., 2023), HiZOO (Zhao et al., 2025b), LOZO (Chen et al., 2025), and PaZO (Zhao et al., 2025a), considering the baselines listed for each model in Tables 1 and 2. All fine-tuning runs use 20,000 training steps and a constant learning rate. For the main results, we evaluate on the validation set every 1000 steps and select the checkpoint with the best validation performance. Hyperparameter settings are provided in Appendix B.

6.1 Masked Language Models

We fine-tune RoBERTa-Large (350M parameters) with $k = 16$ training examples per class on six tasks including SST-2, SST-5, SNLI, MNLI, RTE, and TREC. Table 1 reports mean accuracy and standard deviation over five runs. Zero-shot evaluation and linear probing (LP) (Alain & Bengio, 2016) serve as additional baseline references.

It can be seen from Table 1 that **MSZO improves significantly over single-scale based MeZO on all the six tasks.** Moreover, MSZO outperforms the strong baseline HiZOO (which is a second-order curvature informed method) on the SST-2, SST-5, MNLI, and TREC tasks, and achieves the highest accuracy on these tasks. The largest gain is on MNLI, where MSZO reaches 68.9% accuracy, compared with 58.7% for MeZO and 64.3% for HiZOO, the strongest baseline on this task. On the other five tasks, gains over MeZO range from 1.4 to 3.9 percentage points.

6.2 Autoregressive Language Models

We further evaluate MSZO on three autoregressive models including OPT-1.3B, Qwen3-4B, and Llama3-8B. For each model, we use 1000 training examples per task, retaining the full training set when fewer examples are available, and we evaluate performance on nine downstream tasks. The results are presented in Table 2.

Table 2: Experiments on three autoregressive language models with 1000 training examples (retaining the full training set when fewer examples are available). Average denotes the reported average accuracy (%) over the nine tasks. The best results for each model are highlighted in bold.

Model	Method	SST-2	RTE	CB	BoolQ	WSC	WiC	MultiRC	COPA	ReCoRD	Average
OPT-1.3B	Zero-shot	53.56	53.0	39.3	45.7	43.3	51.5	45.4	75.0	70.5	53.0
OPT-1.3B	ICL	80.0	53.0	46.4	58.7	47.1	51.1	46.2	69.0	71.0	58.1
OPT-1.3B	MeZO	88.5	66.1	67.8	63.4	57.6	60.6	56.3	76.0	**72.3**	67.6
OPT-1.3B	HiZOO	88.5	64.3	67.9	61.4	**62.5**	62.2	59.3	73.0	71.9	67.9
OPT-1.3B	PaZO	89.0	66.4	69.6	63.4	61.5	**63.2**	57.6	75.0	72.1	68.6
OPT-1.3B	MSZO	**92.3**	**67.5**	**78.0**	**66.7**	61.5	62.7	**60.9**	**77.0**	71.4	**70.9**
Qwen3-4B	ICL	88.7	83.5	82.1	**82.7**	60.4	61.5	84.9	76.0	66.8	76.3
Qwen3-4B	MeZO	87.5	83.7	85.7	82.3	61.5	**66.6**	85.0	74.0	69.2	77.3
Qwen3-4B	HiZOO	88.7	81.4	85.3	82.2	63.8	66.4	86.6	74.0	68.0	77.4
Qwen3-4B	LOZO	86.6	80.1	85.7	82.2	62.0	65.9	85.2	75.0	68.7	76.8
Qwen3-4B	MSZO	**91.9**	**83.8**	**87.5**	80.1	**68.2**	64.2	**86.6**	**78.0**	**69.5**	**78.9**
Llama3-8B	MeZO	92.2	74.4	69.6	76.4	63.5	57.8	77.6	88.0	**85.6**	76.1
Llama3-8B	HiZOO	93.5	75.1	68.8	76.8	63.5	**59.7**	78.2	89.0	**85.6**	76.7
Llama3-8B	LOZO	93.6	74.7	69.6	77.1	**63.9**	58.6	78.1	90.0	84.4	76.7
Llama3-8B	MSZO	**93.9**	**78.7**	**75.0**	**81.3**	63.6	58.5	**80.1**	**92.0**	84.9	**78.7**

For OPT-1.3B, the results in Table 2 show that MSZO achieves the highest average accuracy of 70.9%, **outperforming the second-best method PaZO by 2.3%**. It obtains the best results on six of the nine tasks, including SST-2, RTE, CB, BoolQ, MultiRC, and COPA. The largest improvement occurs on CB, where MSZO exceeds PaZO by 8.4%. The gains also extend to the BoolQ and MultiRC tasks, on which MSZO outperforms all the compared baselines. Its advantage over MeZO is maintained when compared with Hessian informed methods HiZOO and PaZO, which incorporate second-order curvature information into ZO optimization.

The advantage of MSZO also extends to Qwen3-4B, where it achieves the best average accuracy and **matches or exceeds the strongest baseline on seven of the nine tasks**. The results on SST-2 and COPA show that the other evaluated ZO baselines do not improve upon in-context learning on these two tasks, whereas MSZO surpasses in-context learning on both the tasks. MSZO also obtains a clear improvement on WSC, exceeding the second-best ZO baseline HiZOO by 4.4%. On the whole, the overall benefit of the proposed schedule extends to Qwen3-4B.

With the larger Llama3-8B model, MSZO achieves the best accuracy on six tasks and again achieves the highest average accuracy of 78.7%, **outperforming both the strong baselines HiZOO and LOZO by 2.0%**. The improvements are most pronounced on RTE, CB, and BoolQ, where MSZO consistently exceeds the evaluated ZO baselines. It also achieves the best results on SST-2, MultiRC, and COPA. On the WSC, WiC, and ReCoRD tasks, MSZO does not attain the best performance, thus the increase in average accuracy does not imply an advantage on every task. Nevertheless, the superior average performance on all the three models demonstrates the effectiveness of the proposed scheduled multi-scale method across the evaluated autoregressive models.

6.3 Memory Overhead and Wall-clock Runtime

Memory overhead comparison. We measure peak GPU memory on MultiRC, with an average input length of 400 tokens, using OPT-1.3B, Qwen3-4B, and Llama3-8B on an NVIDIA A100. As shown in Table 3, MSZO matches the reported peak memory of MeZO and zero-shot inference across all three models. These results indicate that the additional perturbation scale introduces no additional peak-memory overhead in the evaluated settings.

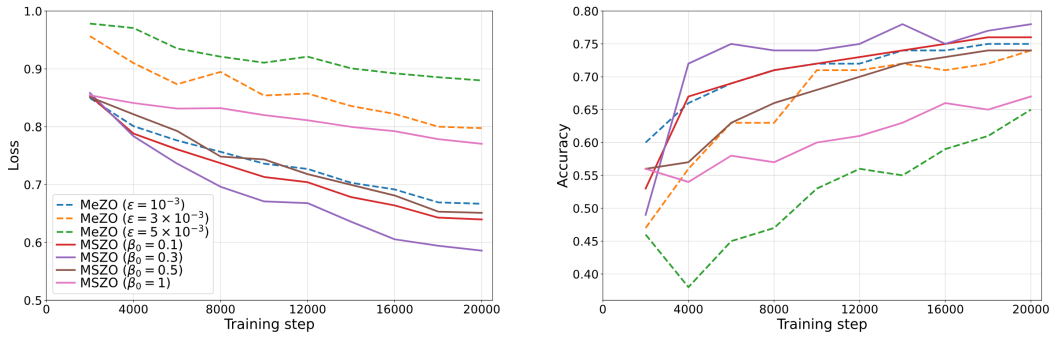
Wall-clock runtime comparison. Table 4 reports the average wall-clock time per training step on SST-2. MSZO takes approximately $1.6\times$ as long as MeZO across the four models. This overhead is consistent with the additional forward evaluations: MSZO uses two evaluations per step in the initial single-scale stage and four in the subsequent multi-scale stage. Its accuracy gains at a fixed training-step budget therefore come at the cost of additional computation.

Table 3: Peak memory on the MultiRC task (average 400 tokens) with three different models on an NVIDIA A100.

Method	OPT-1.3B	Qwen3-4B	Llama3-8B
Zero-shot	4 GB	10 GB	17 GB
MeZO	4 GB	10 GB	17 GB
HiZOO	7 GB	18 GB	41 GB
PaZO	9 GB	21 GB	36 GB
MSZO	4 GB	10 GB	17 GB

Table 4: Wall-clock time per training step on SST-2, averaged over the full training run. The batch size is 64 for RoBERTa-Large and 16 for the other models.

Model	MeZO	HiZOO	MSZO
RoBERTa-Large	0.21 s	0.30 s	0.34 s
OPT-1.3B	0.26 s	0.38 s	0.42 s
Qwen3-4B	0.64 s	0.89 s	1.02 s
Llama3-8B	0.75 s	1.11 s	1.20 s



(a) Training loss

(b) Test accuracy

Figure 2: Sensitivity to the perturbation scale ϵ and initial mixing coefficient β_0 on CB using OPT-1.3B. Dashed lines represent MeZO with different perturbation scales. Solid lines represent MSZO with different initial mixing coefficients.

6.4 Hyperparameter Ablation

We examine sensitivity to the perturbation scale and initial mixing coefficient on CB using OPT-1.3B, with 20,000 training steps and a learning rate of 2×10^{-7}. We compare $\epsilon \in \{10^{-3}, 3 \times 10^{-3}, 5 \times 10^{-3}\}$ for MeZO and $\beta_0 \in \{0.1, 0.3, 0.5, 1\}$ for MSZO. Figure 2 reports training loss and test accuracy. Note that MeZO ($\epsilon = 10^{-3}$) is equivalent to MSZO with $\beta_0 = 0$.

Perturbation scale. The smallest tested scale yields the best final loss and accuracy for MeZO, e.g., $\epsilon = 10^{-3}$. Increasing the fixed perturbation scale degrades both metrics, suggesting that a larger scale alone is insufficient to explain the gains of MSZO.

Mixing coefficient. MSZO performs best with a moderate initial weight (e.g., $\beta_0 = 0.3$), while larger weights weaken its advantage. With $\beta_0 = 1$, the second stage starts with only the large-scale estimator ($\epsilon_l = 5 \times 10^{-3}$) and gradually shifts toward the small-scale estimator as β_t decays to zero. It can be seen from Figure 2 that starting the second stage with $\beta_0 = 1$ performs significantly worse than using intermediate initial mixing weights. This suggests that retaining a small-scale contribution at the start of the second stage is beneficial.

7 Conclusion

In this work, we identified a stage-dependent perturbation-scale effect of ZO gradient estimates in fine-tuning LLMs. Motivated by this finding, we proposed a scheduled multi-scale ZO optimizer MSZO, which starts with a small perturbation scale and subsequently combines two scales through a scheduled mixing weight in the second stage. Experiments across four models and diverse tasks show that MSZO improves average downstream performance over strong single-scale ZO baselines. These gains require additional forward evaluations in the second stage while preserving the low memory footprint of ZO optimization methods. Our findings highlight perturbation-scale scheduling as a useful component of memory-efficient LLM fine-tuning.

AI USE STATEMENT

We used generative AI tools to assist with polishing the manuscript, including improving its clarity, grammar, and organization. All AI-assisted text was reviewed and edited by the authors to ensure accuracy and consistency with the intended meaning. The authors 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.

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. In *Advances in Neural Information Processing Systems*, 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.

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.

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.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yicheng Lang, Changsheng Wang, Yihua Zhang, Mingyi Hong, Zheng Zhang, Wotao Yin, and Sijia Liu. Powering up zeroth-order training via subspace gradient orthogonalization. *arXiv preprint arXiv:2602.17155*, 2026.

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. In *Forty-third International Conference on Machine Learning*, 2026.

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

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*, 2019b.

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.

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.

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

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 *Proceedings of the Forty-third International Conference on Machine Learning*, 2026.

Yao Shu, Zhongxiang Dai, Weicong Sng, Arun Verma, Patrick Jaillet, and Bryan Kian Hsiang Low. Zeroth-order optimization with trajectory-informed derivative estimation. In *The Eleventh International Conference on Learning Representations*, 2023.

James C Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins apl technical digest*, 19(4):482–492, 1998.

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.

Xiaoxing Wang, Wenxuan Guo, Jianlin Su, Xiaokang Yang, and Junchi Yan. Zarts: On zero-order optimization for neural architecture search. *Advances in Neural Information Processing Systems*, 35:12868–12880, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 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.

Junming Zhang, Shuyu Yin, Peilin Liu, Rendong Ying, and Fei Wen. Curvature-aware zeroth-order optimization for memory-efficient test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 836–846, 2026.

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.

Liang Zhang, Bingcong Li, Kiran Koshy Thekumparampil, Sewoong Oh, and Niao He. Dpzero: Private fine-tuning of language models without backpropagation. In *Forty-first International Conference on Machine Learning*, 2024a.

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*, 2022a.

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*, 2024b.

Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jinfeng Yi, Mingyi Hong, Shiyu Chang, and Sijia Liu. How to robustify black-box ml models? a zeroth-order optimization perspective. In *International Conference on Learning Representations*, 2022b.

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.

A THE PROOF OF CONVERGENCE ANALYSIS

A.1 PROOF OF LEMMA 1

Proof. For $\epsilon > 0$, define the Gaussian-smoothed objective as

$$\mathcal{L}_\epsilon(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{z}}\left[\mathcal{L}(\boldsymbol{\theta} + \epsilon \boldsymbol{z})\right]. \tag{18}$$

By Gaussian Stein identity and symmetry of the Gaussian Distribution,

$$\begin{aligned}
\mathbb{E}_{\mathcal{B},\boldsymbol{z}}\left[d_{\epsilon,\mathcal{B}}(\boldsymbol{\theta},\boldsymbol{z})\boldsymbol{z}\right] &= \mathbb{E}_{\mathcal{B},\boldsymbol{z}}\left[\frac{\mathcal{L}(\boldsymbol{\theta}+\epsilon\boldsymbol{z};\mathcal{B}) - \mathcal{L}(\boldsymbol{\theta}-\epsilon\boldsymbol{z};\mathcal{B})}{2\epsilon}\boldsymbol{z}\right] \\
&= \mathbb{E}_{\mathcal{B},\boldsymbol{z}}\left[\frac{\mathcal{L}(\boldsymbol{\theta}+\epsilon\boldsymbol{z};\mathcal{B})}{2\epsilon}\boldsymbol{z}\right] - \mathbb{E}_{\mathcal{B},\boldsymbol{z}}\left[\frac{\mathcal{L}(\boldsymbol{\theta}-\epsilon\boldsymbol{z};\mathcal{B})}{2\epsilon}\boldsymbol{z}\right] \\
&= \frac{1}{2}\mathbb{E}_{\mathcal{B},\boldsymbol{z}}\left[\nabla\mathcal{L}(\boldsymbol{\theta}+\epsilon\boldsymbol{z};\mathcal{B})\right] + \frac{1}{2}\mathbb{E}_{\mathcal{B},\boldsymbol{z}}\left[\nabla\mathcal{L}(\boldsymbol{\theta}-\epsilon\boldsymbol{z};\mathcal{B})\right] \\
&= \nabla\mathcal{L}_\epsilon(\boldsymbol{\theta}).
\end{aligned} \tag{19}$$

Therefore,

$$\mathbb{E}_{\mathcal{B},\boldsymbol{z}}[\widehat{\boldsymbol{g}}_t] = (1-\beta_t)\nabla\mathcal{L}_{\epsilon_s}(\boldsymbol{\theta}) + \beta_t\nabla\mathcal{L}_{\epsilon_l}(\boldsymbol{\theta}). \tag{20}$$

Since $\mathcal{L}$ is L-smooth, we have

$$\begin{aligned}
\|\nabla\mathcal{L}_\epsilon(\boldsymbol{\theta}) - \nabla\mathcal{L}(\boldsymbol{\theta})\| &\le \mathbb{E}_{\boldsymbol{z}}\left[\|\nabla\mathcal{L}(\boldsymbol{\theta}+\epsilon\boldsymbol{z}) - \nabla\mathcal{L}(\boldsymbol{\theta})\|\right] \\
&\le L\epsilon\mathbb{E}\|\boldsymbol{z}\| \\
&\le L\epsilon\sqrt{d}.
\end{aligned} \tag{21}$$

Using the convex combination above gives

$$\begin{aligned}
\|\mathbb{E}[\widehat{\boldsymbol{g}}_t] - \nabla\mathcal{L}(\boldsymbol{\theta})\| &= \|(1-\beta_t)\nabla\mathcal{L}_{\epsilon_s}(\boldsymbol{\theta}) - (1-\beta_t)\nabla\mathcal{L}(\boldsymbol{\theta}) + \beta_t\nabla\mathcal{L}_{\epsilon_l}(\boldsymbol{\theta}) - \beta_t\nabla\mathcal{L}(\boldsymbol{\theta})\| \\
&\le (1-\beta_t)\|\nabla\mathcal{L}_{\epsilon_s}(\boldsymbol{\theta}) - \nabla\mathcal{L}(\boldsymbol{\theta})\| + \beta_t\|\nabla\mathcal{L}_{\epsilon_l}(\boldsymbol{\theta}) - \nabla\mathcal{L}(\boldsymbol{\theta})\| \\
&\le L\sqrt{d}\left[(1-\beta_t)\epsilon_s + \beta_t\epsilon_l\right] \\
&\le L\sqrt{d}\left[(1-\beta_0)\epsilon_s + \beta_0\epsilon_l\right] \\
&= L\sqrt{d}\bar{\epsilon}_\beta,
\end{aligned} \tag{22}$$

where $\bar{\epsilon}_\beta = (1-\beta_0)\epsilon_s + \beta_0\epsilon_l$. This proves the bias bounds.

For the second-moment bound, first consider the estimator constructed from the global loss $\mathcal{L}$. Let

$$q(\boldsymbol{z}) = (1-\beta_t)d_{\epsilon_s}(\boldsymbol{\theta},\boldsymbol{z}) + \beta_t d_{\epsilon_l}(\boldsymbol{\theta},\boldsymbol{z}) - \nabla\mathcal{L}(\boldsymbol{\theta})^\top\boldsymbol{z}. \tag{23}$$

Then, we have

$$\widehat{\boldsymbol{g}}_t = \boldsymbol{z}\boldsymbol{z}^\top\nabla\mathcal{L}(\boldsymbol{\theta}) + q(\boldsymbol{z})\boldsymbol{z}. \tag{24}$$

We obtain Second Moment by

$$\begin{aligned}
\mathbb{E}\|\widehat{\boldsymbol{g}}_t\|^2 &= \mathbb{E}\left[\left(\boldsymbol{z}\boldsymbol{z}^\top\nabla\mathcal{L}(\boldsymbol{\theta}) + q(\boldsymbol{z})\boldsymbol{z}\right)^T\left(\boldsymbol{z}\boldsymbol{z}^\top\nabla\mathcal{L}(\boldsymbol{\theta}) + q(\boldsymbol{z})\boldsymbol{z}\right)\right] \\
&= \mathbb{E}\|\boldsymbol{z}\boldsymbol{z}^\top\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 + 2\mathbb{E}[q(\boldsymbol{z})\langle\boldsymbol{z}\boldsymbol{z}^\top\nabla\mathcal{L}(\boldsymbol{\theta}),\boldsymbol{z}\rangle] + \mathbb{E}[q(\boldsymbol{z})^2\|\boldsymbol{z}\|^2] \\
&\le 2\mathbb{E}\|\boldsymbol{z}\boldsymbol{z}^\top\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 + 2\mathbb{E}[q(\boldsymbol{z})^2\|\boldsymbol{z}\|^2]
\end{aligned} \tag{25}$$

The L-smoothness of loss function implies

$$\begin{aligned}
\|\nabla_{\boldsymbol{z}}q(\boldsymbol{z})\| &= \|(1-\beta_t)\nabla\mathcal{L}(\boldsymbol{\theta}+\epsilon_s\boldsymbol{z}) - (1-\beta_t)\nabla\mathcal{L}(\boldsymbol{\theta}) + \beta_t\nabla\mathcal{L}(\boldsymbol{\theta}-\epsilon_l\boldsymbol{z}) - \beta_t\nabla\mathcal{L}(\boldsymbol{\theta})\| \\
&\le (1-\beta_t)\|\nabla\mathcal{L}(\boldsymbol{\theta}+\epsilon_s\boldsymbol{z}) - \nabla\mathcal{L}(\boldsymbol{\theta})\| + \beta_t\|\nabla\mathcal{L}(\boldsymbol{\theta}-\epsilon_l\boldsymbol{z}) - \nabla\mathcal{L}(\boldsymbol{\theta})\| \\
&\le L\|z\|\left[(1-\beta_t)\epsilon_s + \beta_t\epsilon_l\right] \\
&\le L\bar{\epsilon}_\beta\|\boldsymbol{z}\|.
\end{aligned} \tag{26}$$

For $q(-\boldsymbol{z}) = -q(\boldsymbol{z})$, $\mathbb{E}\left(q(\boldsymbol{z})\right) = 0, \mathrm{Var}\left[q(\boldsymbol{z})\right] = \mathbb{E}\left[q(\boldsymbol{z})^2\right]$. Using the Gaussian Poincaré inequality, it follows that

$$\begin{aligned}\mathbb{E}\left[q(\boldsymbol{z})^2\right] &= \mathrm{Var}\left[q(\boldsymbol{z})\right] \\ &\leq \mathbb{E}\left[\| \nabla_z q(\boldsymbol{z}) \|^2\right] \\ &\leq L^2 \bar{\epsilon}_\beta^2 \mathbb{E}\|\boldsymbol{z}\|^2 \\ &= L^2 \bar{\epsilon}_\beta^2 d.\end{aligned} \tag{27}$$

Using Gaussian integration by parts,

$$\begin{aligned}\mathbb{E}\left[q(\boldsymbol{z})^2\|\boldsymbol{z}\|^2\right] &= d\mathbb{E}[q(\boldsymbol{z})^2] + 2\mathbb{E}[q(\boldsymbol{z})\boldsymbol{z}^\top \nabla q(\boldsymbol{z})] \\ &\leq d\mathbb{E}[q(\boldsymbol{z})^2] + 2\mathbb{E}[|q(\boldsymbol{z})|\|\boldsymbol{z}\|\|\nabla q(\boldsymbol{z})\|] \\ &\leq d\mathbb{E}[q(\boldsymbol{z})^2] + \frac{1}{2}\mathbb{E}[q(\boldsymbol{z})^2\|\boldsymbol{z}\|^2] + 2\mathbb{E}\|\nabla q(\boldsymbol{z})\|^2\end{aligned} \tag{28}$$

Thus,

$$\begin{aligned}\mathbb{E}[q(\boldsymbol{z})^2\|\boldsymbol{z}\|^2] &\leq 2d\mathbb{E}[q(\boldsymbol{z})^2] + 4\mathbb{E}\|\nabla q(\boldsymbol{z})\|^2 \\ &\leq 2(d+2)\mathbb{E}\|\nabla q(\boldsymbol{z})\|^2 \\ &\leq 2L^2\bar{\epsilon}_\beta^2 d(d+2).\end{aligned} \tag{29}$$

Since

$$\mathbb{E}\left[\|\boldsymbol{z}\boldsymbol{z}^\top \nabla \mathcal{L}(\boldsymbol{\theta})\|^2\right] = (d+2)\|\nabla \mathcal{L}(\boldsymbol{\theta})\|^2, \tag{30}$$

we have

$$\mathbb{E}\left[\|\widehat{\boldsymbol{g}}_t\|^2\right] \leq 2(d+2)\|\nabla \mathcal{L}(\boldsymbol{\theta})\|^2 + 4L^2 d(d+2)\bar{\epsilon}_\beta^2. \tag{31}$$

□

A.2 Proof of Theorem 1

Proof. From the standard descent lemma, it follows that

$$\mathcal{L}(\boldsymbol{\theta_{t+1}}) \leq \mathcal{L}(\boldsymbol{\theta_t}) + \nabla \mathcal{L}(\boldsymbol{\theta_t})^\top (\boldsymbol{\theta_{t+1}} - \boldsymbol{\theta_t}) + \frac{L}{2}\|\boldsymbol{\theta_{t+1}} - \boldsymbol{\theta_t}\|^2 \tag{32}$$

Substitute the MSZO parameter update $\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} - \eta \widehat{\boldsymbol{g}}_t$ into the descent lemma,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta_{t+1}}) &\leq \mathcal{L}(\boldsymbol{\theta_t}) - \eta \nabla \mathcal{L}(\boldsymbol{\theta_t})^\top \widehat{\boldsymbol{g}}_t + \frac{L\eta^2}{2}\|\widehat{\boldsymbol{g}}_t\|^2 \\ &= \mathcal{L}(\boldsymbol{\theta_t}) - \eta \nabla \mathcal{L}(\boldsymbol{\theta_t})^\top \left(\widehat{\boldsymbol{g}}_t - \nabla \mathcal{L}(\boldsymbol{\theta_t}) + \nabla \mathcal{L}(\boldsymbol{\theta_t})\right) + \frac{L\eta^2}{2}\|\widehat{\boldsymbol{g}}_t\|^2 \\ &= \mathcal{L}(\boldsymbol{\theta_t}) - \eta \nabla \mathcal{L}(\boldsymbol{\theta_t})^\top \left(\widehat{\boldsymbol{g}}_t - \nabla \mathcal{L}(\boldsymbol{\theta_t})\right) - \eta\|\nabla \mathcal{L}(\boldsymbol{\theta_t})\|^2 + \frac{L\eta^2}{2}\|\widehat{\boldsymbol{g}}_t\|^2 \\ &\leq \mathcal{L}(\boldsymbol{\theta_t}) - \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{\theta_t})\|^2 + \frac{\eta}{2}\|\widehat{\boldsymbol{g}}_t - \nabla \mathcal{L}(\boldsymbol{\theta_t})\|^2 + \frac{L\eta^2}{2}\|\widehat{\boldsymbol{g}}_t\|^2\end{aligned} \tag{33}$$

Taking the conditional expectation with respect to the mini-batch $\mathcal{B}_t$ and the Gaussian direction $\boldsymbol{z}_t$, conditioned on $\boldsymbol{\theta}_t$, gives

$$\mathbb{E}_t\left[\mathcal{L}(\boldsymbol{\theta}_{t+1})\right] \leq \mathcal{L}(\boldsymbol{\theta}_t) - \eta \left\langle \nabla \mathcal{L}(\boldsymbol{\theta}_t), \mathbb{E}_t[\widehat{\boldsymbol{g}}_t] \right\rangle + \frac{L\eta^2}{2}\mathbb{E}_t\|\widehat{\boldsymbol{g}}_t\|^2, \tag{34}$$

where $\mathbb{E}_t[\cdot]$ denotes the conditional expectation given $\boldsymbol{\theta}_t$.

Define the bias

$$\boldsymbol{b}_t = \mathbb{E}_t[\widehat{\boldsymbol{g}}_t] - \nabla \mathcal{L}(\boldsymbol{\theta}_t). \tag{35}$$

Then

$$-\left\langle \nabla \mathcal{L}(\boldsymbol{\theta}_t), \mathbb{E}_t[\widehat{\boldsymbol{g}}_t] \right\rangle = -\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2 - \left\langle \nabla \mathcal{L}(\boldsymbol{\theta}_t), \boldsymbol{b}_t \right\rangle. \tag{36}$$

By Young's inequality,

$$-\left\langle \nabla \mathcal{L}(\boldsymbol{\theta}_t), \boldsymbol{b}_t \right\rangle \leq \frac{1}{2}\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{2}\|\boldsymbol{b}_t\|^2. \tag{37}$$

According to Lemma 1,

$$\|\boldsymbol{b}_t\|^2 \leq L^2 d\bar{\epsilon}_\beta^2. \tag{38}$$

Therefore,

$$-\langle \nabla\mathcal{L}(\boldsymbol{\theta}_t), \mathbb{E}_t[\widehat{\boldsymbol{g}}_t]\rangle \leq -\frac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{L^2 d}{2}\bar{\epsilon}_\beta^2. \tag{39}$$

Combining (34), (39), and Lemma 1, we obtain

$$\begin{aligned}\mathbb{E}_t\left[\mathcal{L}(\boldsymbol{\theta}_{t+1})\right] \leq \mathcal{L}(\boldsymbol{\theta}_t) - \frac{\eta}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\eta L^2 d}{2}\bar{\epsilon}_\beta^2 \\ + \frac{L\eta^2}{2}\left[2(d+2)\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + 4L^2 d(d+2)\bar{\epsilon}_\beta^2 + d(d+2)\sigma_B^2\right].\end{aligned} \tag{40}$$

Rearranging the terms yields

$$\begin{aligned}\mathbb{E}_t\left[\mathcal{L}(\boldsymbol{\theta}_{t+1})\right] \leq \mathcal{L}(\boldsymbol{\theta}_t) - \eta\left[\frac{1}{2} - L\eta(d+2)\right]\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 \\ + \eta L^2 d\left[\frac{1}{2} + 2L\eta(d+2)\right]\bar{\epsilon}_\beta^2 + \frac{L\eta^2}{2}d(d+2)\sigma_B^2.\end{aligned} \tag{41}$$

Taking the total expectation and summing (41) over $t = 0, \ldots, T-1$ give

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_T)] \leq \mathcal{L}(\boldsymbol{\theta}_0) - \eta\left[\frac{1}{2} - L\eta(d+2)\right]\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 \\ + T\eta L^2 d\left[\frac{1}{2} + 2L\eta(d+2)\right]\bar{\epsilon}_\beta^2 + \frac{L\eta^2 T}{2}d(d+2)\sigma_B^2.\end{aligned} \tag{42}$$

Let

$$\Delta_0 = \mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}(\boldsymbol{\theta}^\star), \tag{43}$$

where $\mathcal{L}(\boldsymbol{\theta}^\star)$ is a lower bound of the objective. For

$$0 < \eta < \frac{1}{2L(d+2)}, \tag{44}$$

we have $\frac{1}{2} - L\eta(d+2) > 0$. Hence,

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 \leq \frac{\Delta_0}{\eta T\left[\frac{1}{2} - L\eta(d+2)\right]} \\ + \frac{L^2 d\left[\frac{1}{2} + 2L\eta(d+2)\right]}{\frac{1}{2} - L\eta(d+2)}\bar{\epsilon}_\beta^2 \\ + \frac{L\eta d(d+2)}{2\left[\frac{1}{2} - L\eta(d+2)\right]}\sigma_B^2.\end{aligned} \tag{45}$$

To optimize the convergence bound with respect to the learning rate, we choose the optimal learning rate that minimizes the derived upper bound is

$$\eta^\star = \frac{1}{2L(d+2)\left[1 + \sqrt{1 + \frac{dT\left(3L^2\bar{\epsilon}_\beta^2 + \sigma_B^2/2\right)}{4L(d+2)\Delta_0}}\right]}. \tag{46}$$

Substituting $\eta^\star$ into (45), we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 \leq L^2 d\bar{\epsilon}_\beta^2 + \frac{8L(d+2)\Delta_0}{T} \\ + 2\sqrt{\frac{4L(d+2)\Delta_0}{T}\left[\frac{4L(d+2)\Delta_0}{T} + 3L^2 d\bar{\epsilon}_\beta^2 + \frac{d\sigma_B^2}{2}\right]}.\end{aligned} \tag{47}$$

This completes the proof. □

Table 5: 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}
	ϵ	1e-3
	Weight Decay	0
HiZOO	Batch Size	64
	Learning Rate	{1e-7, 1e-6, 1e-5}
	ϵ	1e-3
	Weight Decay	0
LOZO	Batch Size	64
	Learning Rate	{1e-7, 1e-6, 1e-5}
	ϵ	1e-3
	Rank(r)	{1, 2, 4}
	Interval(v)	{50, 100}
	Weight Decay	0
MSZO	Batch Size	64
	Learning Rate	{1e-7, 5e-7, 1e-6}
	ϵ_s	1e-3
	ϵ_l	5e-3
	β_0	0.2
	T_1	$0.2T$
	Weight Decay	0

B Experiment Setup

B.1 Dataset

For RoBERTa-Large, we consider classification datasets including SST-2, SST-5, MNLI, SNLI, RTE and TREC. For OPT-1.3B, Qwen3-4B and Llama3-8B, we consider the datasets including SST-2, RTE, CB, BoolQ, WSC, WIC, MultiRC, Copa, and ReCoRD.

B.2 Hyperparameter Setting

We use the hyperparameter settings in Tables 5 to 7 for experiments on RoBERTa-Large, OPT 1.3B, Qwen3-4B 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.

C DETAILS OF MSZO

We present the detailed MSZO algorithm in Algorithm 1.

Table 6: 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}
	ϵ	1e-3
HiZOO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	ϵ	1e-3
PaZO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	ϵ	1e-3
MSZO	Batch Size	16
	Learning Rate	{1e-7, 2e-7, 5e-7, 1e-6}
	ϵ_s	1e-3
	ϵ_l	5e-3
	β_0	0.2
	T_1	$0.2T$

Table 7: Hyperparameter setting used for Qwen3-4B and Llama3-8B experiments. All experiments use a constant learning rate schedule.

Experiment	Hyperparameters	Values
MeZO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	ϵ	1e-3
HiZOO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	ϵ	1e-3
LOZO	Batch Size	16
	Learning Rate	{1e-7, 5e-7, 1e-6}
	ϵ	1e-3
	Rank(r)	{1, 2, 4}
	Interval(v)	{50, 100}
MSZO	Batch Size	16
	Learning Rate	{1e-7, 2e-7, 5e-7, 1e-6}
	ϵ_s	1e-3
	ϵ_l	5e-3
	β_0	0.2
	T_1	$0.2T$

Algorithm 1 Scheduled Multi-Scale Zeroth-Order Optimization (MSZO)

Require: Parameters $\boldsymbol{\theta} \in \mathbb{R}^d$, loss $\mathcal{L}$, dataset $\mathcal{D}$, training steps T, learning rate η, perturbation scales $0 < \epsilon_s < \epsilon_l$, transition iteration $T_1 \in \{1, \ldots, T-1\}$, initial mixing coefficient $\beta_0 \in [0, 1]$

for $t = 0, \ldots, T-1$ **do**
  Sample a mini-batch $\mathcal{B}_t$ and a random seed s
  if $t < T_1$ **then**
    $\beta_t \leftarrow 0$
  else
    $\beta_t \leftarrow \beta_0 \dfrac{T-t}{T-T_1}$
  end if
  $\boldsymbol{\theta} \leftarrow \textsc{PerturbParameters}(\boldsymbol{\theta}, \epsilon_s, s)$
  $\ell_s^+ \leftarrow \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}_t)$
  $\boldsymbol{\theta} \leftarrow \textsc{PerturbParameters}(\boldsymbol{\theta}, -2\epsilon_s, s)$
  $\ell_s^- \leftarrow \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}_t)$
  $\boldsymbol{\theta} \leftarrow \textsc{PerturbParameters}(\boldsymbol{\theta}, \epsilon_s, s)$ ▷ Restore parameters
  $d_{\epsilon_s} \leftarrow (\ell_s^+ - \ell_s^-)/(2\epsilon_s)$
  $d_t \leftarrow d_{\epsilon_s}$
  if $t \geq T_1$ **then**
    $\boldsymbol{\theta} \leftarrow \textsc{PerturbParameters}(\boldsymbol{\theta}, \epsilon_l, s)$ ▷ Reuse the same direction
    $\ell_l^+ \leftarrow \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}_t)$
    $\boldsymbol{\theta} \leftarrow \textsc{PerturbParameters}(\boldsymbol{\theta}, -2\epsilon_l, s)$
    $\ell_l^- \leftarrow \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}_t)$
    $\boldsymbol{\theta} \leftarrow \textsc{PerturbParameters}(\boldsymbol{\theta}, \epsilon_l, s)$ ▷ Restore parameters
    $d_{\epsilon_l} \leftarrow (\ell_l^+ - \ell_l^-)/(2\epsilon_l)$
    $d_t \leftarrow (1-\beta_t) d_{\epsilon_s} + \beta_t d_{\epsilon_l}$
  end if
  Reset the random number generator with seed s
  for $i = 1, \ldots, d$ **do**
    Sample $z_i \sim \mathcal{N}(0, 1)$
    $\theta_i \leftarrow \theta_i - \eta d_t z_i$
  end for
end for
return $\boldsymbol{\theta}$
function $\textsc{PerturbParameters}(\boldsymbol{\theta}, \delta, s)$
  Reset the random number generator with seed s
  for $i = 1, \ldots, d$ **do**
    Sample $z_i \sim \mathcal{N}(0, 1)$
    $\theta_i \leftarrow \theta_i + \delta z_i$
  end for
  return $\boldsymbol{\theta}$
end function