ADAPTIVE ZEROTH-ORDER OPTIMIZATION VIA SAMPLE-WISE SECOND MOMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Zeroth-order optimization (ZO) estimates gradients using function evaluations without requiring access to first-order derivatives. Existing ZO methods either rely on isotropic perturbations or require additional function evaluations to obtain adaptive curvature information, introducing extra computation overhead and slowing the convergence. We propose two adaptive ZO optimizers that share an efficient diagonal second-moment estimator built from per-sample gradient surrogates. This state shapes both perturbations and updates, with distinct update scalings and shared adaptive damping for stability and perturbation-scale control. We provide convergence proofs for both variants and demonstrate the first variant's superior performance on test functions with heterogeneous curvature. Experiments on both full-model and parameter-efficient language-model fine-tuning show that standard and adaptive ZO baselines require up to $3.3\times$ as many forward passes as our methods, while our methods achieve competitive final performance.

1 INTRODUCTION

Adaptive optimization methods play a central role in modern machine learning by adjusting parameter updates to the structure of the optimization problem. Methods such as AdaGrad (Duchi et al., 2011), Adam (Kingma & Ba, 2015), and AdamW (Loshchilov & Hutter, 2019) use gradient statistics to adapt coordinate-wise step sizes, enabling effective optimization. Beyond these methods, second-order approaches exploit Hessian information (Dauphin et al., 2014; Yao et al., 2021), while natural gradient descent based approaches use the Fisher information matrix or structured approximations (Amari, 1998; Martens & Grosse, 2015) to account for the local geometry of the optimization problem. Together, these approaches highlight the value of efficiently incorporating gradient statistics and curvature information into optimization.

Despite their benefits, these methods require gradients to construct adaptive statistics or curvature approximations. When gradients are unavailable or expensive to compute, zeroth-order optimization offers an alternative by estimating update directions using only function evaluations. This makes it useful for black-box settings (Nikolakakis et al., 2022) and memory-efficient fine-tuning (Malladi et al., 2023). Motivated by the success of adaptive gradient-based optimizers, Chen et al. (2019); Shu et al. (2025) explore Adam-style adaptivity by using moment statistics derived from ZO gradient estimates. Complementing these approaches, HiZOO (Zhao et al., 2025) uses an estimated diagonal Hessian to introduce curvature awareness into the perturbation distribution and to precondition updates. But this curvature awareness is not free: the diagonal Hessian estimation in HiZOO requires an additional forward pass on top of the two already used for central-difference gradient estimation. This leaves open the question of whether effective curvature-aware adaptivity can be achieved in ZO optimization without incurring extra function evaluations.

In this work, we exploit the sample-wise decomposition of the ZO gradient estimator to efficiently extract a diagonal, sample-wise second-moment curvature signal from the same forward passes used for gradient estimation. To the best of our knowledge, this is the first use of this decomposition to extract sample-wise second-moment information for curvature-aware ZO. By separating sample-specific finite-difference coefficients from the shared perturbation direction, we compute the average squared per-sample gradient estimates without explicitly constructing or storing the individual per-sample gradient estimates. We further introduce adaptive damping to stabilize the resulting preconditioner and control the effective finite-difference perturbation magnitude. Integrating these

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.

components, we propose two adaptive ZO variants that share the same curvature estimation and damping mechanism but differ in how they scale the gradient update. Specifically, by viewing the perturbation as the product of a random vector and a perturbation scale, the first variant preconditions the random vector, while the second adapts the perturbation scale independently for each parameter. We highlight the superiority of the first variant in cases of high heterogeneity of loss curvature on test functions. Furthermore, we show the efficiency of both methods in language-model finetuning of OPT-1.3B (Zhang et al., 2022) and Llama3-3.2 (Touvron et al., 2023), where HiZOO can reach up to $2.8\times$ forward passes compared to ours. Our main contributions are as follows:

- We derive a computationally efficient formulation for extracting diagonal second-moment information from sample-wise ZO gradient surrogates in a mini-batch fashion, requiring neither additional function evaluations nor explicit construction or storage of per-sample gradient vectors.
- We combine the proposed curvature estimator with adaptive damping to develop two optimization variants with distinct update-scaling mechanisms.
- Experiments on both synthetic objectives and language-model fine-tuning show competitive predictive performance while substantially improving query efficiency over existing ZO methods, with support for both full-parameter and LoRA fine-tuning. We complement these results with a convergence analysis both variants establishing convergence to stationarity under the stated assumptions.

2 Related Work

Zeroth-order optimization. Classical stochastic ZO methods construct gradient estimators using randomly perturbed objective evaluations and provide convergence guarantees for convex and non-convex optimization (Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017). More recently, ZO optimization has gained renewed attention as a memory-efficient alternative to backpropagation for large neural networks. MeZO (Malladi et al., 2023) adapts the classical two-point ZO estimator to operate in-place and reconstruct perturbations from random seeds, enabling large-model fine-tuning with memory usage close to inference. Subsequent benchmarking work Zhang et al. (2024) evaluated ZO on LLM-finetuning schemes. Additionally, sparsity in ZO training is explored in Chen et al. (2024); Liu et al. (2026) and ZO finetuning for quantized networks is investigated in (Feng et al., 2024; Shang et al., 2026).

Adaptive zeroth-order optimization. Several works extend adaptive first-order optimization mechanisms to the gradient-free setting. ZO-AdaMM (Chen et al., 2019) maintains first- and second-moment statistics of ZO gradient estimates to construct coordinate-wise adaptive updates. ZO-AdaMU (Jiang et al., 2024) instead incorporates momentum and uncertainty directly into the perturbation mechanism, improving the stability and convergence of ZO optimization for large-model fine-tuning. R-AdaZO (Shu et al., 2025) further revisits adaptive moment estimation in ZO optimization and constructs refined first- and second-moment estimators to reduce the effect of the high variance inherent in random-direction gradients. DiZO (Tan et al., 2026) learns layer-wise projections for ZO fine-tuning of the pretrained Query and Value weights; however, this mechanism is specifically tailored to transformer-based language models. In contrast to Adam-style ZO approaches Chen et al. (2019); Jiang et al. (2024); Shu et al. (2025), our methods do not use or save additional first-moment momentum and utilize the squared per-sample directional derivatives to preserve variation lost through mini-batch averaging and to adapt the perturbation geometry.

Curvature- and second-order-aware zeroth-order optimization. A complementary line of work incorporates explicit curvature information into ZO optimization. More recent methods target memory-efficient neural-network adaptation: HiZOO (Zhao et al., 2025) estimates diagonal Hessian information to precondition ZO fine-tuning. ZOHA (Ye et al., 2025) develops Hessian-aware ZO estimators for improved convergence in a manner similar to (Zhao et al., 2025). PaZO (Zhao et al., 2026) further provides a theoretical motivation for preconditioning in ZO optimization, showing that vanilla ZO-SGD can fail to attain the ideal convergence rate and that an appropriately chosen preconditioner can improve this rate. LOREN (Seung et al., 2026) learns a low-rank curvature representation for anisotropic perturbations using a REINFORCE leave-one-out estimator with multiple

perturbations per step. In contrast to methods that explicitly approximate the Hessian or learn a separate curvature representation, our method exploits the structure of standard ZO evaluations to construct a sample-wise gradient second-moment surrogate directly from per-sample finite differences, which admits an empirical-Fisher-like interpretation.

3 METHODS

In this section, we introduce two variants that use the same online estimate of the diagonal sample-wise second-moment. In the first variant, this estimate is used to precondition the perturbation vector, whereas in the second, it determines an adaptive perturbation scale. Let $\mathcal{L}$ denote the loss, $\mathcal{B}$ denote the mini-batch, and $\mathbf{A}_t$ denote our diagonal matrix. The first variant of the two-sided ZO estimator is

$$\widehat{\mathbf{g}}_t = \frac{\mathcal{L}\left(\boldsymbol{\theta}_t + \epsilon \mathbf{A}_t \mathbf{z}_t; \mathcal{B}\right) - \mathcal{L}\left(\boldsymbol{\theta}_t - \epsilon \mathbf{A}_t \mathbf{z}_t; \mathcal{B}\right)}{2\epsilon} \mathbf{A}_t \mathbf{z}_t, \tag{1}$$

where $\epsilon > 0$ denotes the nominal perturbation scale and $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a random perturbation direction. The second variant is define as

$$\widehat{\mathbf{g}}_t^* = \frac{\mathcal{L}\left(\boldsymbol{\theta}_t + \epsilon \mathbf{A}_t \mathbf{z}_t; \mathcal{B}\right) - \mathcal{L}\left(\boldsymbol{\theta}_t - \epsilon \mathbf{A}_t \mathbf{z}_t; \mathcal{B}\right)}{2\epsilon \mathbf{A}_t} \mathbf{z}_t \tag{2}$$

For both cases, setting $\mathbf{A}_t = \mathbf{I}$ recovers a vanilla ZO-SGD, in which perturbations are isotropic and all parameter directions use the same scale. In particular, $(\mathbf{A}_t)$ introduces parameter-dependent scaling based on the estimated curvature, assigning smaller perturbations to directions with larger curvature estimates and thereby inducing an adaptive anisotropic perturbation geometry. We define $\mathbf{A}_t = (\boldsymbol{\Sigma}_t + \lambda_t \mathbf{I})^{-1/2}$, where $\boldsymbol{\Sigma}_t$ indicates the online sample-wise second-moment estimate and λ_t indicates the adaptive damping parameter. In Section 3.1, we introduce the computationally efficient estimation of $\boldsymbol{\Sigma}_\mathbf{t}$, followed by the adaptive damping Section 3.2 and the overall algorithm for the two variants in Section 3.3. Finally, we provide convergence analysis in Section 3.4.

3.1 COMPUTATIONALLY EFFICIENT ESTIMATION OF SECOND-MOMENT

We construct $\boldsymbol{\Sigma}_t$ as a diagonal curvature surrogate whose entries capture parameter-wise gradient second-moment information. A key aspect of our curvature construction is that we estimate the *average of the squared per-sample gradients surrogate*, rather than the square of the mini-batch averaged gradient, which differs from the mini-batch averaged gradient used in Kingma & Ba (2015); Chen et al. (2019). In particular, squaring the mini-batch averaged gradient discards the variation among individual sample gradients and may yield a small value even when the individual gradients have large magnitudes but partially cancel each other. In contrast, averaging the squared per-sample gradients preserves both the mean-gradient contribution and the sample-wise gradient variability, providing a richer measure of parameter-wise gradient magnitude.

Directly obtaining the squared per-sample gradients in first-order optimization can be computationally demanding, since standard mini-batch backpropagation provides only the gradient of the batch-averaged loss rather than the individual per-sample gradients. Computing the latter generally requires per-example backward computations or specialized vectorized gradient implementations, potentially introducing substantial computation and memory overhead for high-dimensional models. We utilize the structure of ZO gradient estimation, which allows a per-sample second moment to be obtained efficiently from the scalar losses already produced by the perturbed forward evaluations.

Consider a mini-batch $\mathcal{B}_t = \{\xi_i\}_{i=1}^B$ of size B, where a common Gaussian direction $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is shared across samples. Consistent with the curvature-preconditioned ZO estimator in Equations 1 and 2, the objective is queried along the transformed direction $\mathbf{A}_t \mathbf{z}_t$. For each sample ξ_i, we therefore define the scalar central finite-difference coefficient as

$$\delta_{t,i} = \frac{\mathcal{L}\left(\boldsymbol{\theta}_t + \epsilon \mathbf{A}_t \mathbf{z}_t; \xi_i\right) - \mathcal{L}\left(\boldsymbol{\theta}_t - \epsilon \mathbf{A}_t \mathbf{z}_t; \xi_i\right)}{2\epsilon}. \tag{3}$$

The coefficients $\delta_{t,i}$ already incorporate the current adaptive perturbation geometry through $\mathbf{A}_t$. We use these coefficients differently for optimization and curvature estimation. For estimating the next

curvature state, we combine each squared coefficient with the element-wise square of the original Gaussian direction,

$$\widehat{\mathbf{s}}_{t,i} = \delta_{t,i}^2 \left(\mathbf{z}_t \odot \mathbf{z}_t\right). \tag{4}$$

This separates the geometry used to query the objective from the coordinate-wise probe used to construct the curvature surrogate. In particular, since $\delta_{t,i}$ already depends on the preconditioned direction $\mathbf{A}_t\mathbf{z}_t$, retaining $\mathbf{z}_t \odot \mathbf{z}_t$ avoids applying the current preconditioner a second time when estimating its successor. The mini-batch curvature statistic then admits the factorized form

$$\mathbf{S}_t = \frac{1}{B}\sum_{i=1}^{B} \widehat{\mathbf{s}}_{t,i} \quad = \left(\frac{1}{B}\sum_{i=1}^{B} \delta_{t,i}^2\right)\left(\mathbf{z}_t \odot \mathbf{z}_t\right). \tag{5}$$

The factorization in eq. (5) avoids explicitly constructing or storing B separate d-dimensional gradient vectors. Instead, the computation requires only the B scalar finite-difference coefficients and the shared d-dimensional perturbation vector. Moreover, these scalar coefficients are obtained directly from the *same perturbed forward evaluations* required by the ZO gradient estimator executed in a mini-batch computation. As a result, the diagonal per-sample second moment can be estimated directly from the same two perturbed forward evaluations used by the ZO update, without requiring additional function queries or per-sample forward passes, making sample-wise curvature information particularly accessible in the ZO setting.

This statistic also admits a curvature-related interpretation. Let $\mathbf{g}_i$ denote the gradient per sample, considering the corresponding full-matrix quantity $\mathbf{G}_t = \frac{1}{B}\sum_{i=1}^{B} \mathbf{g}_i\mathbf{g}_i^\top$, $\mathbf{S}_t$ is a surrogate of $\mathrm{diag}(\mathbf{G}_t)$ in ZO. For negative log-likelihood objectives, $\mathbf{G}_t$ is commonly interpreted as an empirical Fisher matrix Martens (2020) if B is equal to the size of the dataset. The empirical Fisher computed using observed training labels, however, is not in general identical to either the exact Fisher, the generalized Gauss–Newton matrix, or the Hessian. We therefore interpret $\mathbf{S}_t$ primarily as a diagonal per-sample gradient second-moment surrogate, with an empirical-Fisher-like interpretation for negative log-likelihood objectives.

To reduce stochastic variation in the instantaneous estimate $\mathbf{S}_t$, we maintain an exponential moving average (EMA),

$$\mathbf{\Sigma}_t = \rho\mathbf{\Sigma}_t + (1-\rho)\mathbf{S}_t, \qquad 0 \le \rho < 1, \tag{6}$$

where $\mathbf{\Sigma}_1$ is initialized as $\mathbf{I}$. The parameter ρ controls the temporal scale over which the second-moment information is accumulated. A larger ρ produces a smoother but more slowly adapting curvature estimate, whereas a smaller ρ assigns greater weight to the most recent mini-batch.

3.2 Adaptive damping

Damping is commonly used in both first-order adaptive optimizers and adaptive zeroth-order methods by adding a small positive constant to the accumulated second-moment or curvature estimate, primarily to prevent numerical instability when the denominator becomes small. In our setting, the preconditioner play an additional role: it determines not only the scaling of the optimization update, but also the geometry and magnitude of the zeroth-order perturbation itself. Directly using $\mathbf{\Sigma}_t^{-1/2}$ can introduce a significant practical issue: the magnitude of the entries of $\mathbf{\Sigma}_t$ may increase substantially during optimization due to the large magnitude of zeroth-order (ZO) gradients. Because the effective perturbation radius along parameter j is inversely proportional to $\sqrt{\Sigma_{t,j}}$, an overall increase in the scale of $\mathbf{\Sigma}_t$ causes the corresponding perturbation radii to shrink:

$$\Sigma_{t,j} \uparrow \qquad \Longrightarrow \qquad \epsilon_{t,j} = \frac{\epsilon}{\sqrt{\Sigma_{t,j}}} \downarrow. \tag{7}$$

Thus, the magnitude of the curvature estimate directly controls the effective ZO smoothing radius. In particular, a large $\Sigma_{t,j}$ has an effect analogous to reducing the nominal perturbation scale ϵ along parameter j. If the resulting perturbation becomes excessively small, the difference between the two perturbed function evaluations can become correspondingly small, increasing the relative influence of stochastic loss variation and numerical errors on the finite-difference estimate (Liu et al., 2020). More importantly, a simultaneous increase of the entries of $\mathbf{\Sigma}_t$ may undesirably reduce the perturbation scale across the entire parameter space, even when the relative differences between parameter-wise curvature values remain informative.

Algorithm 1 Sample-Wise Informed Zeroth-order Optimization (SWIZO) (SWIZO*)

Require: Initial parameters $\boldsymbol{\theta}_0$, learning rate η, perturbation scale ϵ, EMA coefficient β, number of optimization steps T

Ensure: Optimized parameters $\boldsymbol{\theta}_T$

1: Initialize $\boldsymbol{\Sigma}_1 \leftarrow \mathbf{I}$ and $\lambda_1 \leftarrow 0$

2: **for** $t = 1, \ldots, T$ **do**

3: Sample mini-batch $\mathcal{B}_t = \{\xi_i\}_{i=1}^{B}$

4: Sample $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Random perturbation}

5: $\{\ell_{t,i}^{+}\}_{i=1}^{B} \leftarrow f\Big(\boldsymbol{\theta}_{t-1} + \epsilon\,(\boldsymbol{\Sigma}_t + \lambda_t \mathbf{I})^{-1/2}\,\mathbf{z}_t; \mathcal{B}\Big)$ {Positive perturbation}

6: $\{\ell_{t,i}^{-}\}_{i=1}^{B} \leftarrow f\Big(\boldsymbol{\theta}_{t-1} - \epsilon\,(\boldsymbol{\Sigma}_t + \lambda_t \mathbf{I})^{-1/2}\,\mathbf{z}_t; \mathcal{B}\Big)$ {Negative perturbation}

7: $\delta_{t,i} \leftarrow \dfrac{\ell_{t,i}^{+} - \ell_{t,i}^{-}}{2\epsilon}, \quad \forall i \in \mathcal{B}_t$ {Per-sample finite difference}

8: $s_t \leftarrow \dfrac{1}{B} \displaystyle\sum_{i \in \mathcal{B}_t} \delta_{t,i}^2$ {Mean squared coefficient}

9: $\mathbf{S}_t \leftarrow s_t\,(\mathbf{z}_t \odot \mathbf{z}_t)$ {Second-moment estimate}

10: $\boldsymbol{\Sigma}_t \leftarrow \rho \boldsymbol{\Sigma}_{t-1} + (1-\rho)\mathbf{S}_t$ {EMA curvature update}

11: $\lambda_t \leftarrow 1 - \min_j \Sigma_{t,j}$

12: $\bar{\delta}_t \leftarrow \dfrac{1}{B} \displaystyle\sum_{i \in \mathcal{B}_t} \delta_{t,i}$ {Batch directional derivative}

13: $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta\,\bar{\delta}_t\,(\boldsymbol{\Sigma}_t + \lambda_t \mathbf{I})^{-1/2}\,\mathbf{z}_t$ {Parameter update}

14: $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta\,\bar{\delta}_t\,(\boldsymbol{\Sigma}_t + \lambda_t \mathbf{I})^{1/2}\,\mathbf{z}_t$ {Parameter update}

15: **end for**

16: **return** $\boldsymbol{\theta}_T$

To avoid using the large absolute magnitude of $\boldsymbol{\Sigma}_t$, we adaptively shift it so that its smallest diagonal entry is anchored at one. Specifically, we define an adaptive damping per step λ_t, where

$$\lambda_t = 1 - \min_j \Sigma_{t,j}. \tag{8}$$

This bounds the effective perturbation radius of every parameter direction by the nominal ZO perturbation scale ϵ, $0 < \epsilon(\Sigma_{t,j} + \lambda_t)^{-1/2} \leq \epsilon$, with at least one parameter direction retaining the full perturbation radius ϵ. In particular, our approach removes the large common offset.

This provides two desirable properties. First, the parameter direction with the smallest estimated second moment retains the nominal perturbation scale (ϵ).This prevents all parameter-wise perturbations from shrinking uniformly as the overall magnitude of ($\boldsymbol{\Sigma}_t$) grows. Second, the remaining perturbation radii depend only on their second moments relative to the minimum value. As a result, common shifts in ($\boldsymbol{\Sigma}_t$) do not remove the curvature information nor introduce additional hyperparameters to tune.

3.3 Overall Training Procedure

Algorithm 1 summarizes the complete optimization procedure for both variants. At each iteration, the current second-moment state and adaptive damping determine the parameter-wise perturbation geometry. A random direction is then sampled and used to perform two-sided function evaluations, from which both the ZO update and the per-sample directional loss differences are obtained. The same perturbation geometry is used consistently for the function evaluations and parameter update, while the squared per-sample differences are aggregated to construct the second-moment statistic for the subsequent iteration. The curvature state is updated through an exponential moving average, after which the damping term is recomputed.Afterwards, the parameters are updated either using the update step in line 13 for the first variant (SWIZO) or line 14 for the second one (SWIZO*). In this way, each iteration simultaneously performs optimization and refines the geometry used to generate future ZO queries. We further note that the perturbation vector can be generated parameter-wise in a similar manner to Malladi et al. (2023); Zhao et al. (2025) for memory efficiency.

3.4 Convergence Analysis

We analyze the convergence of the proposed methods under standard assumptions for stochastic non-convex optimization, including L-smoothness and stochastic gradient variance bounded by σ^2 (Assumptions 2 and 3 in Section A), while explicitly accounting for adaptive damping. In particular, we characterize how damping bounds the eigenvalues of the matrix $M_t = (\Sigma_t + \lambda_t I)^{-1}$ and incorporate these bounds into our convergence analysis.

Assumption 1 (Bounds on the adaptive state)**.** *Each diagonal entry of Σ_t lies in the interval $[\alpha_\ell, \alpha_u]$, where $\alpha_\ell \geq 0$. Let $\lambda_t = 1 - \alpha_\ell$ Then each diagonal entry of $(\Sigma_t + \lambda_t I)$ lies in $[1,\ \alpha_u - \alpha_\ell + 1]$. Therefore each diagonal entry of $M_t = (\Sigma_t + \lambda_t I)^{-1}$ lies in $[\beta_\ell, 1]$, where $\beta_\ell \coloneqq \frac{1}{\alpha_u - \alpha_\ell + 1}$.*

Theorem 1. *Define the estimator $g_\epsilon(\theta_t) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathcal{L}(\theta_t + \epsilon M_t^{1/2} z_i) - \mathcal{L}(\theta_t - \epsilon M_t^{1/2} z_i)}{2\epsilon} M_t^{1/2} z_i$, where the update $\theta_{t+1} = \theta_t - \eta_t g_\epsilon(\theta_t)$. Let T be the number of iterations and θ_j given that j sampled uniformly from $\{1, \ldots, T\}$. Choosing $\eta = \frac{1}{8\sqrt{T}\, L \max_t (\, tr(M_t)+2)}$, we obtain the following bound for θ_j, which implies convergence to stationarity as the number of iterations as $T \to \infty$. Detailed proof is presented in Appendix A.*

$$\mathbb{E}\big[\|\nabla \mathcal{L}(\theta_{\mathrm{j}})\|^2\big] \leq \frac{32L(\mathcal{L}(\theta_1) - \mathcal{L}(\theta_*)(\max_t(\mathrm{tr}(M_t) + 2))}{\sqrt{T}\beta_\ell} + \frac{\sigma^2}{\sqrt{T}\beta_\ell} + \mathcal{O}(\epsilon^2) \tag{9}$$

Theorem 2. *For the second variant, let $g_\epsilon^*(\theta_t) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathcal{L}(\theta_t + \epsilon M_t^{1/2} z_i) - \mathcal{L}(\theta_t - \epsilon M_t^{1/2} z_i)}{2\epsilon} M_t^{-1/2} z_i$, where the update $\theta_{t+1} = \theta_t - \eta_t g_\epsilon^*(\theta_t)$. Choosing $\eta = \frac{1}{8\sqrt{T}\, L \max_t (\, tr(M_t^{-1})+2)}$, we obtain the following bound for θ_j, which implies convergence to stationarity as the number of iterations as $T \to \infty$. Detailed proof is presented in Appendix B.*

$$\mathbb{E}\big[\|\nabla \mathcal{L}(\theta_{\mathrm{j}})\|^2\big] \leq \frac{32L(\mathcal{L}(\theta_1) - \mathcal{L}(\theta_*)(\max_t(\mathrm{tr}(M_t^{-1}) + 2))}{\sqrt{T}} + \frac{\sigma^2}{\sqrt{T}} + \mathcal{O}(\epsilon^2). \tag{10}$$

The two bounds eqs. (9) and (10) differ in their dependence on M_t where the first involves $\max_t(\mathrm{tr}(M_t) + 2)/\beta_l$, whereas the second involves $\max_t(\mathrm{tr}(M_t^{-1}) + 2)$.

4 Results

This section presents evaluations on language model fine-tuning tasks and synthetic test functions, comparisons with state-of-the-art ZO methods, and ablation studies examining the contribution of the proposed approach's key components. We compare our methods with three representative ZO baselines: MeZO (Malladi et al., 2023), R-AdaZO (Shu et al., 2025), and HiZOO (Zhao et al., 2025). MeZO (ZO-SGD) serves as the standard memory-efficient ZO baseline for LLM fine-tuning. R-AdaZO represents the recent state of the art in adaptive-moment-based ZO optimization, refining the use of first- and second-moment estimates to improve variance reduction and accelerate convergence. HiZOO represents a curvature-aware baseline similar to our first variant method but estimates curvature diagonal Hessian information through an additional function evaluation. We denote SWIZO as the first variant and SWIZO* for the second one.

4.1 Evaluation on Synthetic functions

We evaluate the methods on two synthetic functions with strongly heterogeneous curvature across parameter dimensions:

$$f_{\mathrm{a}}(x, y) = 10000x^2 + y^2, \tag{11}$$

$$f_{\mathrm{b}}(x, y) = 8(x-1)^2(1.3x^2 + 2x + 1) + 0.5(y-4)^2. \tag{12}$$

We refer to these objectives as *func. (a)* and *func. (b)*, respectively. To simulate noisy function evaluations, we evaluate each objective through

$$\widetilde{f}(x, y; \xi) = f(x, y) \times \max((1 + \sigma_{\mathrm{n}}\xi), 0.05) \qquad \xi \sim \mathcal{N}(0, 1),$$

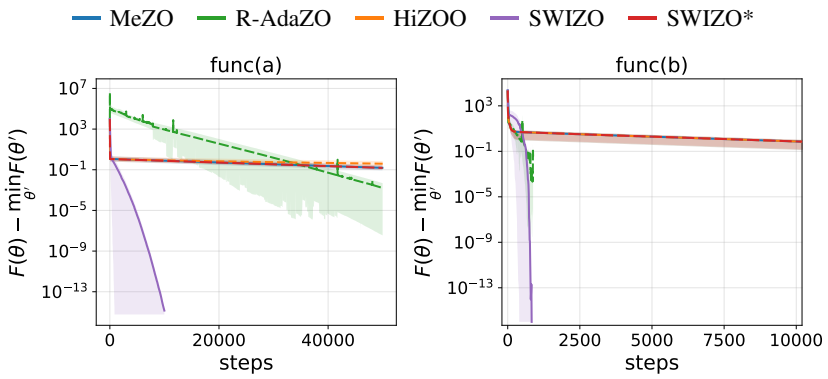

Figure 1: Comparison of ZO methods on test functions. Objective values are plotted on a logarithmic scale against optimization steps. Our approach (SWIZO) converges to the minimum substantially faster than the other methods.

Table 1: Accuracy of fine-tuning on OPT-1.3B and Llama-3.2-3B models.

	OPT-1.3B			Llama-3.2-3B		
Method	**SST-2**	**WiC**	**CB**	**SST-2**	**WiC**	**CB**
MeZO	93.0	59.8	81.0	92.6	61.8	74.0
R-AdaZO	**93.6**	61.2	80.0	91.0	63.6	75.0
HiZOO	93.0	62.0	**88.0**	**93.0**	64.0	**78.0**
SWIZO	**93.6**	**63.4**	84.0	92.8	**64.8**	77.0
SWIZO*	**93.6**	62.0	**88.0**	**93.0**	64.4	75.0

where σ_{n} controls the noise scale and is set to 0.5. Although this noise model is used in the context of CMA-ES (Uchida et al., 2024) rather than two-point ZO optimization, it provides an established setting for evaluation under stochastic function evaluations. This experiment examines whether the optimizer can handle large differences in directional curvature while remaining effective when the available function values are corrupted by noise.

Figure 1 shows that evaluated methods reduce the objective gap below 1 on both functions. For all methods, we terminate optimization once the objective gap reaches 10^{-15} or reach 50K steps. SWIZO converges substantially faster and reaches a considerably smaller objective gap. In particular, on Func. (b), it satisfies this stopping criterion well before exhausting the optimization gap, effectively recovering the optimum within numerical precision. For SWIZO*, ZO-SGD, and HiZOO, they show similar convergence behavior in terms of steps. These results highlight SWIZO's ability to reach the stopping condition efficiently, even when competing methods fail to do so.

4.2 Language Models Finetuning

For language-model fine-tuning, we evaluate our methods on the OPT-1.3B (Zhang et al., 2022) and Llama-3.2-3B (Touvron et al., 2023) models using the SST-2 (Socher et al., 2013), WiC (Pilehvar & Camacho-Collados, 2019), and CB (De Marneffe et al., 2019) datasets. The full-model fine-tuning allows us to evaluate the scalability of the proposed method to a large number of trainable parameters. We further evaluate the case of LoRA (Hu et al., 2022) fine-tuning on OPT-1.3B to demonstrate that our methods is compatible with widely adopted parameter-efficient finetuning techniques. Further experimental setup is detailed in Appendix E.

Accuracy. Table 1 compares the methods on OPT-1.3B and Llama-3.2-3B in full model finetuning. Our variants generally achieve better accuracy than MeZO and R-AdaZO across all evaluated settings. On OPT-1.3B, both variants match the best result on SST-2 and with SWIZO obtaining the highest accuracy on WiC, while SWIZO* also matches the best result on CB, yielding the strongest average performance across the three tasks. On Llama-3.2-3B, SWIZO* matches the best result on

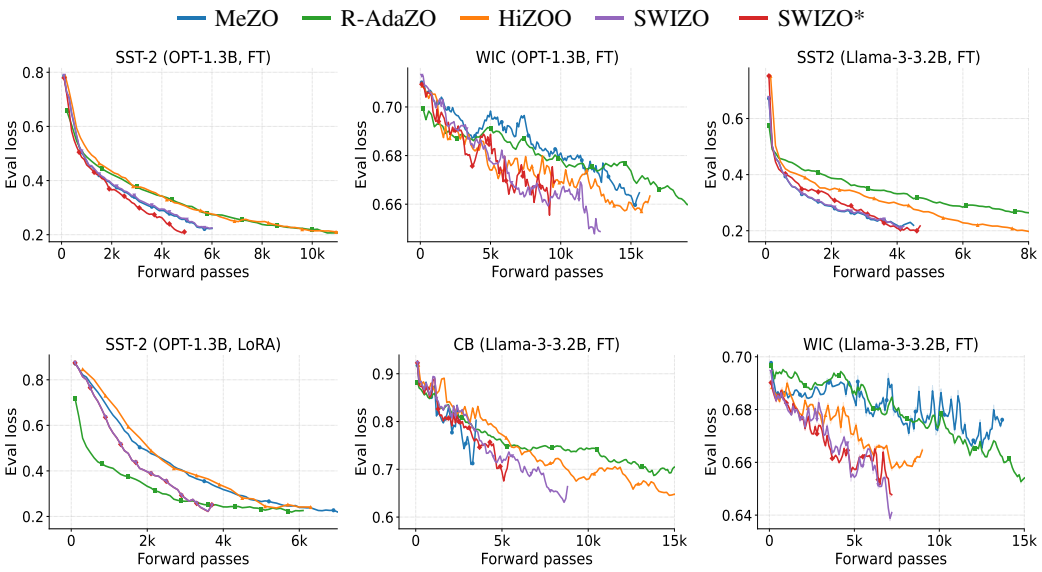

Figure 2: Illustration of test loss between our methods (SWIZO and SWIZO*) and state-of-the-art as a function of the number of forward passes, reflecting convergence relative to computational cost. ZO baselines require up to 3.3× more forward passes as our proposed methods. See also Appendix D for pairwise comparisons on selected examples.

SST-2, whereas SWIZO achieves the highest accuracy on WiC and improves upon SWIZO* on CB. HiZOO achieves the highest accuracy on CB. Overall, both variants remain competitive with explicit Hessian-informed adaptation while relying only on the standard two perturbed forward evaluations.

Furthermore, Table 2 presents the results for LoRA fine-tuning of OPT-1.3B. Overall, the differences among the evaluated methods are smaller in the LoRA setting. Nevertheless, our methods provides a clear improvement on WiC while remaining competitive on the other tasks. Overall, our method consistently improves or maintains competitive accuracy across both full-parameter and LoRA fine-tuning, demonstrating its effectiveness across different adaptation settings.

Forward Passes. We further evaluate the training cost of each method in terms of the number of forward passes required to reach convergence. In Figure 2, we provide the forward passes over the test loss for the considered training examples.

Compared with MeZO, our methods generally require fewer forward passes during full-parameter fine-tuning while also achieving higher accuracy across the evaluated tasks. A similar trend is observed for LoRA fine-tuning on OPT-1.3B, where the adaptive ZO methods converge faster than MeZO, with our methods showing particularly favorable convergence behavior.

Table 2: Accuracy comparison of LoRA fine-tuning on OPT-1.3B for ZO methods.

Method	SST-2	WiC	CB
MeZO	**94.0**	57.0	79.0
R-AdaZO	91.6	58.4	78.0
HiZOO	93.0	57.8	79.0
SWIZO	92.8	**61.2**	79.0
SWIZO*	93.0	60.8	79.0

R-AdaZO remains competitive in the LoRA setting and generally converges faster than MeZO. Its convergence curves typically show rapid initial progress toward a lower training loss, followed by slower improvement in later stages. We attribute this behavior to the dependence of its second-moment estimate on the momentum state, which introduces bias into the adaptive scaling. However, this advantage does not transfer well to the full-parameter fine-tuning setting, where its convergence is comparable to or slower than MeZO and consistently slower than our methods. This slowdown did lead to up to 3.3× more forward passes than those used by our methods (see Appendix D).

HiZOO requires an additional forward pass to estimate the diagonal Hessian, resulting in 3 forward passes per optimization step, compared to the two forward passes for our methods. Moreover, our methods often converge in fewer optimization steps, further reducing the total number of forward passes beyond the per-step savings. Consequently, HiZOO requires up to $2.8\times$ as many forward passes as SWIZO and $2.3\times$ as many as SWIZO* (see Appendix D). Despite the difference in training costs, the two methods generally achieve comparable accuracy across the evaluated tasks.

SWIZO and SWIZO* require similar numbers of forward passes, with SWIZO* generally requiring fewer on OPT-1.3B and SWIZO on Llama-3.2-3B, though the differences are small overall. Overall, our methods combine the query efficiency of two-point ZO optimization with faster and more stable convergence across both full-parameter and LoRA fine-tuning. It generally improves upon MeZO, retains the advantages of adaptivity more consistently than R-AdaZO, and achieves accuracy comparable to HiZOO with substantially fewer forward passes.

Memory overhead. Relative to MeZO, both our methods and HiZOO maintain one additional per-parameter diagonal state, corresponding to the estimated second-moment or Hessian-diagonal information, respectively. R-AdaZO, in contrast, maintains two additional per-parameter states for its first- and second-moment estimates, requiring twice the optimizer-state storage of our methods.

4.3 Ablation

In this section, we discuss the effect of the EMA smoothing parameter ρ on the proposed methods.

Effect of ρ and damping on SWIZO. In Figure 3, we visualize the effect of adaptive damping and $(1-\rho)$ on our approach when using the same learning rate. As shown, the approach exhibits slower convergence without damping. This effect appears more as $(1-\rho)$ increases, i.e., when $\mathbf{\Sigma}_t$ is updated with a higher frequency. This damping effect further provides the advantage of using a higher learning rate with a larger value for $(1-\rho)$.

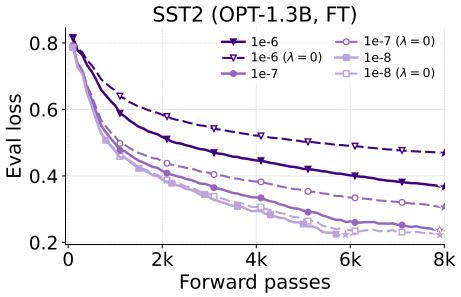

Figure 3: Illustration of the $(1-\rho)$ and adaptive damping on our first variant (SWIZO), where $\lambda = 0$ indicate no adaptive damping.

Effect of ρ on SWIZO*. Figure 4 examines the effect of $1-\rho$ on SST-2 with full-parameter fine-tuning of OPT-1.3B. The tested values yield similar loss trajectories during most of the training, indicating limited sensitivity within this range. Differences emerge toward the end, where smaller values achieve lower evaluation loss, suggesting that stronger smoothing can benefit late-stage convergence. In general, it is less sensitive to the choice of ρ compared to the first variant. An ablation of adaptive damping is provided in Appendix C, showing that the benefits become more visible late in the optimization process.

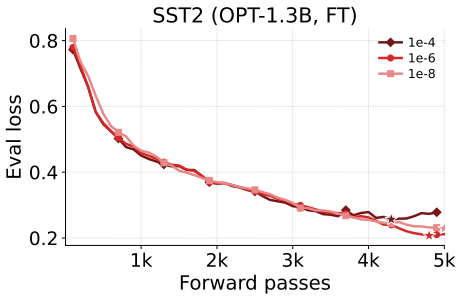

Figure 4: Illustration of $(1-\rho)$ on SWIZO*

5 Conclusion

In this work, we introduced two adaptive zeroth-order optimization variants that combine a diagonal second-moment signal extracted from per-sample loss differences with adaptive damping to guide preconditioning and control the effective perturbation scale. Switching between the variants requires only flipping the sign of the preconditioner's exponent in the final update step of the parameters, making the implementation straightforward. Across language model fine-tuning settings, both variants achieved improved convergence and competitive accuracy with fewer forward passes, highlighting the benefits of adapting perturbations using sample-wise information. The first variant also demonstrated more accurate and substantially faster convergence on test functions with highly heterogeneous curvature.

AI USE STATEMENT

In this work, we used generative AI tools for helping in writing formulation and refinement. We have not used generative AI tools for other tasks with required disclosure are not applicable to this work. Additionally, we used generative AI tools to generate code for plotting figures. We have reviewed the AI-assisted text and checked the generated plotting code and resulting figures. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

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 *International Conference on Learning Representations*, volume 2024, pp. 50185–50206, 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*, volume 32, pp. 7202–7213, 2019.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in neural information processing systems*, 27, 2014.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pp. 107–124, 2019.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Chen Feng, Shaojie Zhuo, Xiaopeng Zhang, Ramchalam K Ramakrishnan, Zhaocong Yuan, and Andrew Z Li. Stepping forward on the last mile. *Advances in Neural Information Processing Systems*, 37:94851–94870, 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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

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.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

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.

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. *Advances in Neural Information Processing Systems*, 38:21139–21169, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Jan R Magnus et al. *The moments of products of quadratic forms in normal variables*. Univ., Instituut voor Actuariaat en Econometrie, 1978.

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.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

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

Konstantinos Nikolakakis, Farzin Haddadpour, Dionysis Kalogerias, and Amin Karbasi. Black-box generalization: Stability of zeroth-order learning. *Advances in neural information processing systems*, 35:31525–31541, 2022.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1267–1273, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1128. URL https://aclanthology.org/N19-1128.

Hyunseok Seung, Jaewoo Lee, and Hyunsuk Ko. Low-rank curvature for zeroth-order optimization in llm fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25235–25242, 2026.

Sifeng Shang, Jiayi Zhou, Chenyu Lin, Minxian Li, and Kaiyang Zhou. Fine-tuning quantized neural networks with zeroth-order optimization. In *International Conference on Learning Representations*, volume 2026, pp. 128868–128885, 2026.

Yao Shu, Qixin Zhang, Kun He, and Zhongxiang Dai. Refining adaptive zeroth-order optimization at ease. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55403–55426. PMLR, 2025.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Qitao Tan, Jun Liu, Zheng Zhan, Caiwen Ding, Yanzhi Wang, Xiaolong Ma, Jaewoo Lee, Jin Lu, and Geng Yuan. Harmony in divergence: Towards fast, accurate, and memory-efficient zeroth-order llm fine-tuning. *Advances in Neural Information Processing Systems*, 38:175003–175037, 2026.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Kento Uchida, Kenta Nishihara, and Shinichi Shirakawa. Cma-es with adaptive reevaluation for multiplicative noise. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 731–739, 2024.

Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10665–10673, 2021.

Haishan Ye, Zhichao Huang, Cong Fang, Chris Junchi Li, and Tong Zhang. Hessian-aware zeroth-order optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(6):4869–4877, June 2025. ISSN 0162-8828. doi: 10.1109/TPAMI.2025.3548810. URL https://doi.org/10.1109/TPAMI.2025.3548810.

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 D. 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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 59173–59190. PMLR, 2024.

Hanzhen Zhao, Ding Shihong, Cong Fang, and Zhouchen Lin. Pazo: Preconditioned accelerated zeroth-order optimization for fine-tuning llms. *Advances in Neural Information Processing Systems*, 38:111163–111195, 2026.

Yanjun Zhao, Sizhe Dang, Haishan Ye, Guang Dai, Yi Qian, and Ivor Tsang. Second-order fine-tuning without pain for llms: A hessian informed zeroth-order optimizer. In *International Conference on Learning Representations*, volume 2025, pp. 43496–43520, 2025.

A PROOF OF THEOREM 1

This proof follows the analysis of Zhao et al. (2025), with modifications to account for our adaptive preconditioner and damping mechanism.

Let $M_t \coloneqq (\Sigma_t + \lambda_t I)^{-1}$ For any vector v, define $\|v\|_{M_t}^2 \coloneqq v^\top M_t v$.

Assumption 2 (Smoothness)**.** *The objective L is L-smooth. That is, for any $\theta_1, \theta_2 \in \mathbb{R}^d$,*

$$\mathcal{L}(\theta_2) \le \mathcal{L}(\theta_1) + \langle \nabla \mathcal{L}(\theta_1), \theta_2 - \theta_1 \rangle + \frac{L}{2} \|\theta_2 - \theta_1\|^2 .$$

Assumption 3 (Stochastic-gradient variance)**.** $\mathbb{E}\left[\|\nabla \mathcal{L}(\theta; \xi) - \nabla \mathcal{L}(\theta)\|^2\right] \le \sigma^2$.

We restate Theorem 1 here. Define the zeroth-order estimator

$$g_\epsilon(\theta_t) = \frac{1}{n} \sum_{i=1}^{n} \frac{\mathcal{L}(\theta_t + \epsilon M_t^{1/2} z_i) - \mathcal{L}(\theta_t - \epsilon M_t^{1/2} z_i)}{2\epsilon} M_t^{1/2} z_i. \tag{13}$$

Let T be the number of iterations and θ_j given that j sampled uniformly from $\{1, \dots, T\}$. Choosing $\eta = \frac{1}{8\sqrt{T}\, L \max_t (\, tr(M_t)+2)}$, we obtain the following bound for θ_j, which implies convergence to stationary as the number of iterations as $T \to \infty$.

$$\mathbb{E}\big[\|\nabla \mathcal{L}(\theta_{\mathrm{j}})\|^2\big] \le \frac{32L(\mathcal{L}(\theta_1) - \mathcal{L}(\theta_*)(\max_t(\mathrm{tr}(M_t) + 2))}{\sqrt{T}\beta_\ell} + \frac{\sigma^2}{\sqrt{T}\beta_\ell} + \mathcal{O}(\epsilon^2) \tag{14}$$

Proof.

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\theta_{t+1}) \mid \theta_t] &\le \mathcal{L}(\theta_t) - \eta_t \mathbb{E}[\langle \nabla \mathcal{L}(\theta_t), g_\epsilon(\theta_t) \rangle] + \frac{L\eta_t^2}{2} \mathbb{E}\Big[\|g_\epsilon(\theta_t)\|^2\Big] \\
&\le \mathcal{L}(\theta_t) - \eta_t \|\nabla \mathcal{L}(\theta_t)\|_{M_t}^2 + \eta_t \mathcal{O}(\epsilon \|\nabla \mathcal{L}(\theta_t)\|) \\
&\quad + 2\eta_t^2 L \left(\mathrm{tr}(M_t) + 2\right) \|\nabla \mathcal{L}(\theta_t)\|_{M_t}^2 \\
&\quad + 2\eta_t^2 L \left(\mathrm{tr}(M_t) + 2\right) \sigma^2 + \mathcal{O}(\epsilon^2) \\
&\le \mathcal{L}(\theta_t) - \frac{\eta_t}{2} \|\nabla \mathcal{L}(\theta_t)\|_{M_t}^2 \\
&\quad + 2\eta_t^2 L \left(\mathrm{tr}(M_t) + 2\right) \|\nabla \mathcal{L}(\theta_t)\|_{M_t}^2 \\
&\quad + 2\eta_t^2 L \left(\mathrm{tr}(M_t) + 2\right) \sigma^2 + \mathcal{O}(\epsilon^2) \\
&= \mathcal{L}(\theta_t) - \frac{\eta_t}{2} \left[1 - 4\eta_t L \left(\mathrm{tr}(M_t) + 2\right)\right] \|\nabla \mathcal{L}(\theta_t)\|_{M_t}^2 \\
&\quad + 2\eta_t^2 L \left(\mathrm{tr}(M_t) + 2\right) \sigma^2 + \mathcal{O}(\epsilon^2) \\
&\le \mathcal{L}(\theta_t) - \frac{\eta_t}{4} \|\nabla \mathcal{L}(\theta_t)\|_{M_t}^2 + 2\eta_t^2 L \left(\mathrm{tr}(M_t) + 2\right) \sigma^2 + \mathcal{O}(\epsilon^2).
\end{aligned}$$

where the first inequality is from Assumption 2 and the update $\theta_{t+1} = \theta_t - \eta_t g_\epsilon(\theta_t)$, the second inequality is due to Lemmas 1 and 3 and the last inequality is due to the appropriate choice of η_t. Rearranging the one-step descent inequality and summing over $t = 1, \dots, T$ yields

$$\begin{aligned}
\frac{\eta}{4} \sum_{t=1}^{T} \mathbb{E}\big[\|\nabla \mathcal{L}(\theta_t)\|_{M_t}^2\big] &\le \sum_{t=1}^{T} \mathbb{E}[\mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1})] + 2\eta^2 L\sigma^2 \sum_{t=1}^{T} (\mathrm{tr}(M_t) + 2) + \mathcal{O}(T\epsilon^2) \\
&\le \mathcal{L}(\theta_1) - \mathcal{L}(\theta_*) + 2T\eta^2 L\sigma^2 \max_t (\mathrm{tr}(M_t) + 2) + \mathcal{O}(T\epsilon^2).
\end{aligned} \tag{15}$$

Let $\theta_{out} = \theta_j$, where j is sampled uniformly from $\{1, \dots, T\}$. Since $M_t \succeq \beta_\ell I$, we obtain

$$\begin{aligned}\mathbb{E}\big[\|\nabla\mathcal{L}(\theta_{\text{out}})\|^2\big] &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla\mathcal{L}(\theta_t)\|^2\big] \\ &\le \frac{1}{T\beta_\ell}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla\mathcal{L}(\theta_t)\|^2_{M_t}\big] \\ &\le \frac{4(\mathcal{L}(\theta_1)-\mathcal{L}(\theta_*))}{T\eta\beta_\ell} + \frac{8\eta L\max_t(\mathrm{tr}(M_t)+2)}{\beta_\ell}\sigma^2 + \mathcal{O}(\epsilon^2) \qquad (16)\end{aligned}$$

Choosing $\eta = \frac{1}{8\sqrt{T}\,L\max_t(\,tr(M_t)+2)}$ gives

$$\mathbb{E}\big[\|\nabla\mathcal{L}(\theta_{\text{j}})\|^2\big] \le \frac{32L(\mathcal{L}(\theta_1)-\mathcal{L}(\theta_*)(\max_t(\mathrm{tr}(M_t)+2))}{\sqrt{T}\beta_\ell} + \frac{\sigma^2}{\sqrt{T}\beta_\ell} + \mathcal{O}(\epsilon^2) \qquad (17)$$

□

Lemma 1 (Zhao et al. (2025)). *The estimator in equation 13 satisfies*

$$\mathbb{E}[g_\epsilon(\theta_t)] = M_t\nabla\mathcal{L}(\theta_t) + \mathcal{O}(\epsilon).$$

Lemma 2 ((Magnus et al., 1978)). *Let A and B be symmetric matrices and let $z \sim \mathcal{N}(0, I_d)$. Then*

$$\mathbb{E}\big[(z^\top Az)(z^\top Bz)\big] = (\mathrm{tr}\,A)(\mathrm{tr}\,B) + 2\,\mathrm{tr}(AB).$$

Lemma 3. *The following result follows the analysis of Zhao et al. (2025, Lemma B.4), with a corrected constant in the β_u term, diagonal entry of $M_t = (\Sigma_t + \lambda_t I)^{-1}$ lies in $[\beta_\ell, \beta_u]$ and $\beta_u = 1$ due to the damping. The estimator in equation 13 satisfies*

$$\begin{aligned}\mathbb{E}\Big[\|g_\epsilon(\theta_t)\|^2\Big] &\le 4\,(\mathrm{tr}(M_t)+2)\,\|\nabla\mathcal{L}(\theta_t)\|^2_{M_t} \\ &\quad + 4\beta_u\,(\mathrm{tr}(M_t)+2)\,\sigma^2 + \mathcal{O}(\epsilon^2). \qquad (18)\end{aligned}$$

Proof. Using the expansion from Zhao et al. (2025),

$$\begin{aligned}\mathbb{E}\Big[\|g_\epsilon(\theta_t)\|^2\Big] &= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}M_t^{1/2}z_iz_i^\top M_t^{1/2}\nabla\mathcal{L}(\theta_t) + \mathcal{O}(\epsilon)\right\|^2\right] \\ &\le 2\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}M_t^{1/2}z_iz_i^\top M_t^{1/2}\nabla\mathcal{L}(\theta_t)\right\|^2\right] + \mathcal{O}(\epsilon^2) \\ &\le \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left\|M_t^{1/2}z_iz_i^\top M_t^{1/2}\nabla\mathcal{L}(\theta_t)\right\|^2\right] + \mathcal{O}(\epsilon^2).\end{aligned}$$

Let $A = M_t, \qquad B = M_t^{1/2}\nabla\mathcal{L}(\theta_t)\nabla^\top\mathcal{L}(\theta_t)M_t^{1/2}$. Then

$$\begin{aligned}\mathrm{tr}(A)\,\mathrm{tr}(B) &= \mathrm{tr}(M_t)\nabla^\top\mathcal{L}(\theta_t)M_t\nabla\mathcal{L}(\theta_t), \\ \mathrm{tr}(AB) &= \nabla^\top\mathcal{L}(\theta_t)M_t^2\nabla\mathcal{L}(\theta_t).\end{aligned}$$

Therefore, by Lemma 2,

$$\begin{aligned}\mathbb{E}\Big[\|g_\epsilon(\theta_t)\|^2\Big] &\le 2\,\mathrm{tr}(M_t)\nabla^\top\mathcal{L}(\theta_t)M_t\nabla\mathcal{L}(\theta_t) \\ &\quad + 4\nabla^\top\mathcal{L}(\theta_t)M_t^2\nabla\mathcal{L}(\theta_t) + \mathcal{O}(\epsilon^2). \\ &\le (2\,\mathrm{tr}(M_t) + 4\beta_u)\,\|\nabla\mathcal{L}(\theta_t)\|^2_{M_t} + \mathcal{O}(\epsilon^2).\end{aligned}$$

and for a stochastic gradient, following Zhao et al. (2025), we have

$$\mathbb{E}\Big[\|\nabla\mathcal{L}(\theta_t)\|^2_{M_t}\Big] \le 2\,\|\nabla\mathcal{L}(\theta_t)\|^2_{M_t} + 2\beta_u\sigma^2.$$

Combining the last two displays yields equation 18. □

B PROOF OF THEOREM 2

We restate the theorem again. Given the estimator

$$g_\epsilon^*(\theta_t) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathcal{L}(\theta_t + \epsilon M_t^{1/2} z_i) - \mathcal{L}(\theta_t - \epsilon M_t^{1/2} z_i)}{2\epsilon} M_t^{-1/2} z_i. \tag{19}$$

where the update $\theta_{t+1} = \theta_t - \eta_t g_\epsilon^*(\theta_t)$. Let T be the number of iterations and θ_j given that j sampled uniformly from $\{1, \ldots, T\}$. Choosing $\eta = \frac{1}{8\sqrt{T}\, L \max_t (\, tr(M_t^{-1})+2)}$

$$\mathbb{E}\big[\|\nabla \mathcal{L}(\theta_{\mathrm{j}})\|^2\big] \leq \frac{32L(\mathcal{L}(\theta_1) - \mathcal{L}(\theta_*)(\max_t(\mathrm{tr}(M_t^{-1}) + 2))}{\sqrt{T}} + \frac{\sigma^2}{\sqrt{T}} + \mathcal{O}(\epsilon^2). \tag{20}$$

Proof.

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\theta_{t+1}) \mid \theta_t] &\leq \mathcal{L}(\theta_t) - \eta_t \mathbb{E}[\langle \nabla \mathcal{L}(\theta_t), g_\epsilon^*(\theta_t)\rangle] + \frac{L\eta_t^2}{2}\mathbb{E}\Big[\|g_\epsilon^*(\theta_t)\|^2\Big] \\
&\leq \mathcal{L}(\theta_t) - \eta_t \|\nabla \mathcal{L}(\theta_t)\|^2 + \eta_t \mathcal{O}(\epsilon \|\nabla \mathcal{L}(\theta_t)\|) \\
&\quad + 2\eta_t^2 L(\mathrm{tr}(M_t^{-1})(\|\nabla \mathcal{L}(\theta_t)\|_{M_t}^2 + \sigma^2) \\
&\quad + 4\eta_t^2 L(\|\nabla \mathcal{L}(\theta_t)\|^2 + \sigma^2) + \mathcal{O}(\epsilon^2) \\
&\leq \mathcal{L}(\theta_t) - \frac{\eta_t}{2}\|\nabla \mathcal{L}(\theta_t)\|^2 + 2\eta_t^2 L(\mathrm{tr}(M_t^{-1})(\beta_u \|\nabla \mathcal{L}(\theta_t)\|^2 + \sigma^2) \\
&\quad + 4\eta_t^2 L(\|\nabla \mathcal{L}(\theta_t)\|^2 + \sigma^2) + \mathcal{O}(\epsilon^2) \\
&\leq \mathcal{L}(\theta_t) - \frac{\eta_t}{2}\|\nabla \mathcal{L}(\theta_t)\|^2 + \eta_t O(\epsilon \|\nabla \mathcal{L}(\theta_t)\|) \\
&\quad + 2\eta_t^2 L\,\mathrm{tr}(M_t^{-1})\beta_u \|\nabla \mathcal{L}(\theta_t)\|^2 + 2\eta_t^2 L\,\mathrm{tr}(M_t^{-1})\sigma^2 \\
&\quad + 4\eta_t^2 L\|\nabla \mathcal{L}(\theta_t)\|^2 + 4\eta_t^2 L\sigma^2 + O(\epsilon^2) \\
&\leq \mathcal{L}(\theta_t) - \frac{\eta_t}{2}\left[1 - 4L\eta_t^2(\beta_u\,\mathrm{tr}(M_t^{-1}) + 2)\right]\|\nabla \mathcal{L}(\theta_t)\|^2 \\
&\quad + 2L\eta_t^2(\mathrm{tr}(M_t^{-1}) + 2)\sigma^2 + O(\epsilon^2) \\
&\leq \mathcal{L}(\theta_t) - \frac{\eta_t}{4}\|\nabla \mathcal{L}(\theta_t)\|^2 + 2L\eta_t^2((\mathrm{tr}(M_t^{-1}) + 2)\sigma^2 + \mathcal{O}(\epsilon^2).
\end{aligned}$$

where the first inequality is due to Lemma 4 and Lemma 5 and the the last inequality is due to the appropriate choice of η_t and that $\beta_u = 1$. □

$$\begin{aligned}
\frac{\eta}{4}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla \mathcal{L}(\theta_t)\|^2\big] &\leq \sum_{t=1}^{T}\mathbb{E}[\mathcal{L}(\theta_t) - \mathcal{L}(\theta_{t+1})] + 2\eta^2 L\sigma^2 \sum_{t=1}^{T}(\mathrm{tr}(M_t^{-1}) + 2) + \mathcal{O}(T\epsilon^2) \\
&\leq \mathcal{L}(\theta_1) - \mathcal{L}(\theta_*) + 2T\eta^2 L\sigma^2 \max_t(\mathrm{tr}(M_t^{-1}) + 2) + \mathcal{O}(T\epsilon^2).
\end{aligned} \tag{21}$$

Let $\theta_{out} = \theta_j$, where j is sampled uniformly from $\{1, \ldots, T\}$.

$$\begin{aligned}
\mathbb{E}\big[\|\nabla \mathcal{L}(\theta_{\mathrm{out}})\|^2\big] &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla \mathcal{L}(\theta_t)\|^2\big] \\
&\leq \frac{4(\mathcal{L}(\theta_1) - \mathcal{L}(\theta_*))}{T\eta} + 8\eta L \max_t(\mathrm{tr}(M_t^{-1}) + 2)\sigma^2 + \mathcal{O}(\epsilon^2)
\end{aligned} \tag{22}$$

Choosing $\eta = \frac{1}{8\sqrt{T}\, L \max_t (\, tr(M_t^{-1})+2)}$

$$\mathbb{E}\big[\|\nabla \mathcal{L}(\theta_{\mathrm{j}})\|^2\big] \leq \frac{32L(\mathcal{L}(\theta_1) - \mathcal{L}(\theta_*)(\max_t(\mathrm{tr}(M_t^{-1}) + 2))}{\sqrt{T}} + \frac{\sigma^2}{\sqrt{T}} + \mathcal{O}(\epsilon^2). \tag{23}$$

Lemma 4. *The estimator in eq.* (19) *satisfies*

$$\mathbb{E}[g_\epsilon^*(\theta_t)] = \nabla\mathcal{L}(\theta_t) + \mathcal{O}(\epsilon).$$

Proof.

$$\begin{aligned} g_\epsilon^*(\theta_t) &= \frac{1}{n}\sum_{i=1}^{n} \frac{\mathcal{L}(\theta_t + \epsilon M_t^{1/2} z_i) - \mathcal{L}(\theta_t - \epsilon M_t^{1/2} z_i)}{2\epsilon} M_t^{-1/2} z_i \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{2\epsilon \nabla^\top \mathcal{L}(\theta_t) M_t^{1/2} z_i + \mathcal{O}(\epsilon^2)}{2\epsilon} M_t^{-1/2} z_i \\ &= \frac{1}{n}\sum_{i=1}^{n} M_t^{-1/2} z_i z_i^\top M_t^{1/2} \nabla\mathcal{L}(\theta_t) + \mathcal{O}(\epsilon) \end{aligned}$$

Then

$$\mathbb{E}[g_\epsilon^*(\theta_t)] = \nabla\mathcal{L}(\theta_t) + \mathcal{O}(\epsilon).$$

where the second equality is due to Taylor expansion for finite difference Zhao et al. (2025).

□

Lemma 5. *The second moment of eq.* (19)

$$\mathbb{E}\Big[\|g_\epsilon^*(\theta_t)\|^2\Big] \le 2\,\mathrm{tr}(M_t^{-1})(2\,\|\nabla\mathcal{L}(\theta_t)\|_{M_t}^2 + 2\beta_u\sigma^2) + 4(2\,\|\nabla\mathcal{L}(\theta_t)\|^2 + 2\sigma^2) + \mathcal{O}(\epsilon^2)$$

Proof. Using the expansion from the first lemma,

$$\begin{aligned} \mathbb{E}\Big[\|g_\epsilon^*(\theta_t)\|^2\Big] &= \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n} M_t^{-1/2} z_i z_i^\top M_t^{1/2} \nabla\mathcal{L}(\theta_t) + \mathcal{O}(\epsilon)\right\|^2\right] \\ &\le \frac{2}{n}\sum_{i=1}^{n} \mathbb{E}\left[\left\|M_t^{-1/2} z_i z_i^\top M_t^{1/2} \nabla\mathcal{L}(\theta_t)\right\|^2\right] + \mathcal{O}(\epsilon^2) \\ &= 2\,\mathrm{tr}(M_t^{-1})\,\|\nabla\mathcal{L}(\theta_t)\|_{M_t}^2 + 4\,\|\nabla\mathcal{L}(\theta_t)\|^2 + \mathcal{O}(\epsilon^2) \\ &\le 2\,\mathrm{tr}(M_t^{-1})(2\,\|\nabla\mathcal{L}(\theta_t)\|_{M_t}^2 + 2\beta_u\sigma^2) + 4(2\,\|\nabla\mathcal{L}(\theta_t)\|^2 + 2\sigma^2) + \mathcal{O}(\epsilon^2) \end{aligned}$$

where the second equality is achieved by using Lemma 2

$$A = M_t^{-1}, \qquad B = M_t^{1/2}\nabla\mathcal{L}(\theta_t)\nabla^\top\mathcal{L}(\theta_t)M_t^{1/2}.$$

$$\begin{aligned} \mathrm{tr}(A)\,\mathrm{tr}(B) &= \mathrm{tr}(M_t^{-1})\nabla^\top\mathcal{L}(\theta_t)M_t\nabla\mathcal{L}(\theta_t), \\ \mathrm{tr}(AB) &= \nabla^\top\mathcal{L}(\theta_t)\nabla\mathcal{L}(\theta_t) \end{aligned}$$

□

C Further Ablation

Effect of adaptive damping on SWIZO*. We ablate the proposed damping mechanism on SST-2 and WiC using OPT-1.3B as shown in Figure 5. On SST-2, the effect of adaptive damping is relatively small and becomes most visible toward the end of optimization as the value of $\mathbf{\Sigma}$ grows. This behavior is similarly repeated on the WIC dataset.

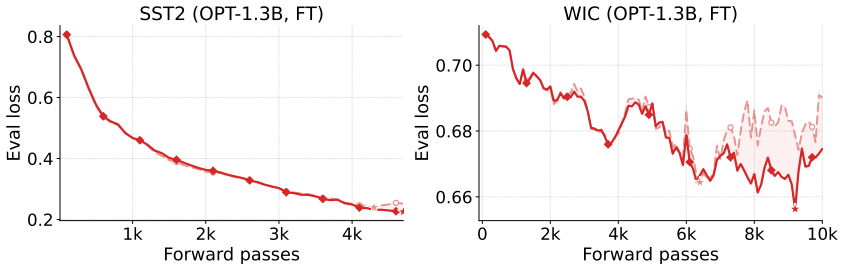

Figure 5: Effect of adaptive damping on our second varient (SWIZO*)

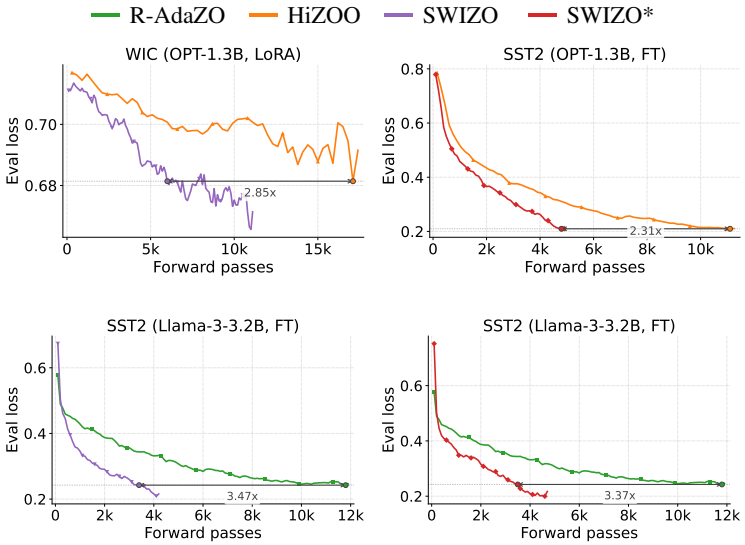

Figure 6: Comparison of evaluation loss between our method and state-of-the-art as a function of the number of forward passes, reflecting convergence relative to computational cost.

D Further visualization

For readability, Figure 6 presents selected examples highlighting substantial gains achieved by our approaches over the baselines.

E Experimental setup

We use a batch size of 16 across all experiments, except for LLaMA on CB, where the batch size is reduced to 4 due to memory limitations. The finite-difference perturbation scale is set to $\epsilon = 10^{-3}$. The remaining hyperparameters used for evaluation are provided in Table 3.

For the two test functions, we initialize (x, y) at $(-1, -1)$ for func(a) and $(7, 7)$ for func(b). We perform a separate grid search to select the hyperparameters of each approach for each test function.

Table 3: Hyperparameter values and ranges used across datasets and models. η denotes the learning rate. FT denotes full-parameter fine-tuning.

Tuning	Method	Learning rate η	Additional hyperparameters
FT	MeZO	$[5 \times 10^{-7},\ 2 \times 10^{-6}]$	—
	R-AdaZO	$[4 \times 10^{-6},\ 8 \times 10^{-6}]$	$\beta_1 = 0.9,\ \beta_2 = 0.999$
	HiZOO	$[10^{-6},\ 2 \times 10^{-6}]$	$\alpha \in [10^{-8}, 10^{-6}]$
	SWIZO	$[10^{-6},\ 2 \times 10^{-6}]$	$1 - \rho \in [10^{-8},\ 10^{-7}]$
	SWIZO*	$[10^{-6},\ 2 \times 10^{-6}]$	$1 - \rho \in [10^{-8},\ 10^{-6}]$
LoRA	MeZO	$[10^{-4},\ 2 \times 10^{-4}]$	—
	R-AdaZO	$[5 \times 10^{-5}, 10^{-4}]$	$\beta_1 = 0.9,\ \beta_2 = 0.999$
	HiZOO	$[8 \times 10^{-5},\ 10^{-4}]$	$\alpha \in [10^{-8}, 10^{-6}]$
	SWIZO	$[8 \times 10^{-5},\ 10^{-4}]$	$1 - \rho \in [10^{-8}, 10^{-7}]$
	SWIZO*	$[8 \times 10^{-5},\ 10^{-4}]$	$1 - \rho \in [10^{-8},\ 10^{-4}]$