SUBZERO+: MEMORY-EFFICIENT ADAPTIVE ZEROTH-ORDER LLM FINE-TUNING IN RANDOM SUBSPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Zeroth-order (ZO) optimization with SGD in random subspaces enables memory-efficient fine-tuning of large language models without backpropagation. However, high gradient estimation noise fundamentally undermines adaptive optimizers like Adam. We propose SubZero+, which achieves practical adaptive ZO optimization through a carefully designed dual low-dimensionality strategy: (i) multi-query forward-difference gradient estimation in periodically refreshed random subspaces to mitigate noise amplification in moment buffers, and (ii) Adam updates with periodic restarts performed directly in low-dimensional space rather than full-parameter space. In experiments, this dual design retains memory overhead comparable to momentum-free ZO methods while achieving stronger optimization performance than the evaluated ZO baselines. Theoretically, in the exact-directional limit, K-query averaging preserves conditional unbiasedness, while the coefficient estimator's covariance and mean-squared error, as well as query-induced second-moment inflation, scale exactly as $1/K$. Extensive experiments across SuperGLUE with models from 1.3B to 32B parameters under both full fine-tuning and LoRA schemes demonstrate consistent improvements over competing ZO methods. SubZero+ significantly narrows the performance gap with first-order optimization while preserving ZO's inference-time memory efficiency. Code is available at https://anonymous.4open.science/r/SubZero-Plus-E8A6.

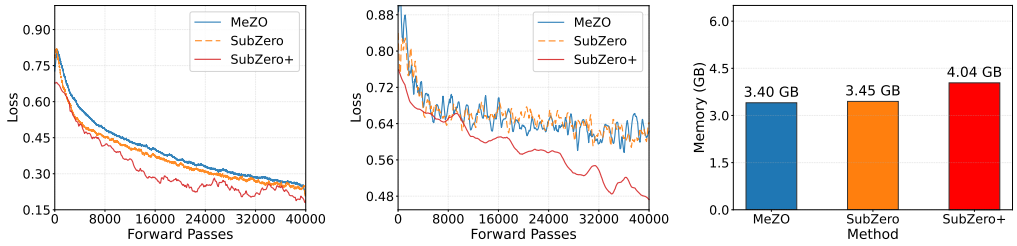
1 INTRODUCTION

Fine-tuning large language models (LLMs) is essential for adapting pretrained models to downstream tasks. Most existing methods rely on backpropagation (BP) to compute first-order (FO) gradients for optimization (Amari, 1993; Kingma & Ba, 2015). As a principled alternative, zeroth-order (ZO) optimization enables fine-tuning without backward passes (Malladi et al., 2023; Zhang et al., 2024; Liu et al., 2025; Yu et al., 2025). By removing BP, ZO methods can significantly reduce activation-memory overhead, simplify the training pipeline, and make fine-tuning practical in scenarios where BP is infeasible or inefficient (Liu et al., 2026). However, in practice it remains challenging to obtain stable and competitive performance with ZO optimization.

A promising direction combines ZO estimation with subspace restriction. In particular, SubZero (Yu et al., 2025) projects ZO gradient estimates onto layer-wise low-dimensional subspaces that are periodically refreshed for both efficiency and effectiveness. This design is motivated by the observation that fine-tuning gradients are often approximately low-rank (Zhang et al., 2023; Malladi et al., 2023; Zhao et al., 2024; Hao et al., 2024). By reducing the effective dimension, subspace methods can improve estimation quality while preserving the inference-level memory benefits of ZO training. Nonetheless, the residual estimation noise poses a particular challenge for adaptive optimizers such as Adam (Kingma & Ba, 2015), since the same noisy estimate is used in both the numerator and the denominator of the parameter update.

A critical observation reveals that despite gradient accumulation across multiple queries, existing methods achieve only marginal improvements in ZO LLM fine-tuning (Peng et al., 2025). We pinpoint a fundamental bottleneck: by performing both gradient estimation and parameter updates in the full-parameter space, existing methods suffer from dimension-dependent variance that dominates even after averaging multiple queries (Duchi et al., 2015; Nesterov & Spokoiny, 2017). Consequently, variance reduction scales only as $O(1/K)$ for K queries, while the dimension-dependent noise dominates. Furthermore, adaptive optimizers crucially depend on the stability of gradient directions

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.



(a) OPT-1.3B FT on SST-2 (b) LLaMA3.1-8B LoRA on WiC (c) Memory Cost of (a)

Figure 1: Training dynamics and memory cost of SubZero+. (a) Under full-parameter tuning of OPT-1.3B on SST-2, SubZero+ reaches a lower loss than MeZO and SubZero within the same forward-pass budget. (b) Under LoRA tuning of LLaMA3.1-8B on WiC, SubZero+ gives a smoother trajectory and reaches a lower loss. (c) For the setting of (a), SubZero+ introduces only $\sim$0.6GB additional GPU memory overhead compared with MeZO and SubZero.

across iterations. In high-dimensional settings, ZO gradients tend to exhibit weaker directional correlation (Ghadimi & Lan, 2013), so momentum accumulation fails to yield meaningful acceleration. Together, these factors explain why Adam does not consistently outperform SGD in ZO LLM fine-tuning (Zhang et al., 2024; Guo et al., 2025; Shu et al., 2025), even though it can be more effective in FO training (Glentis et al., 2026).

Contributions. Our main contributions are as follows. Firstly, we propose SubZero+, which features two complementary designs in random subspaces. SubZero+ employs multi-query ZO gradient estimation with the forward-difference scheme, which reduces contamination in the moment estimates without maintaining full-parameter optimizer states. Then it applies Adam directly to the resulting low-dimensional coefficient matrices. SubZero+ stores only two low-dimensional moment matrices per layer and resets them whenever the random bases are refreshed periodically. The former design enhances gradient accumulation by reducing the noise of the gradient estimator, thereby improving the quality of the moment estimates. The latter design performs momentum accumulation in the low-dimensional space, enabling Adam to produce stable and efficient updates while keeping the optimizer state lightweight. With the dual low-dimensionality, SubZero+ achieves both lower memory consumption and improved optimization performance.

Secondly, we characterize the query-induced noise that enters SubZero+'s coefficient-space moments in the exact-directional limit. Theorem 1 shows that the K-query coefficient estimator is conditionally unbiased and that its covariance and mean-squared error decrease exactly as $1/K$. Theorem 2 further shows that each coordinate's expected squared write contains an additive global-energy term suppressed by exactly $1/K$. These conditional, single-step identities clarify why multiple queries are especially relevant to the inputs of Adam's accumulators.

Finally, we evaluate SubZero+ on SuperGLUE across model sizes from 1.3B to 32B under both full-parameter tuning and LoRA (Hu et al., 2022). Across all settings, SubZero+ consistently outperforms the evaluated ZO baselines, narrowing the performance gap to FO methods while introducing minimal additional memory overhead. These results indicate that multi-query estimation and compact moments in random low-rank subspaces provide a practical pathway for adaptive ZO LLM fine-tuning.

2 Related Work

ZO Gradient Estimation. ZO optimization replaces BP with finite-difference gradient estimates that rely only on forward passes. MeZO (Malladi et al., 2023) pioneered SPSA-based (Spall, 1992) ZO fine-tuning for LLMs, achieving inference-level memory usage while exhibiting gradient estimation variance that scales with the parameter dimension. Subsequent work reduces this variance through sparse perturbations (Liu et al., 2025; Guo et al., 2025) and low-rank subspace projections (Chen et al., 2025; Yu et al., 2025). LOZO (Chen et al., 2025) factorizes perturbations as low-rank matrices, while SubZero (Yu et al., 2025) enforces column-orthonormal projection matrices via QR decomposition. All these methods use a single query per iteration, which limits the achievable variance reduction.

ZO Parameter Update. Many ZO methods, including MeZO (Malladi et al., 2023) and SubZero (Yu et al., 2025), apply vanilla SGD to the estimated gradients. LOZO (Chen et al., 2025) performs SGDM

within a low-rank subspace and requires negligible additional memory. TeZO (Sun et al., 2025) further exploits temporal low-rank structure across gradients and can be extended to an Adam variant. However, previous attempts at adaptive ZO optimization have consistently failed to outperform SGD in practice (Malladi et al., 2023; Yu et al., 2025; Guo et al., 2025).

In contrast, adaptive ZO optimization extends Adam/RMSProp-style adaptive normalization to the ZO setting. ZO-AdaMM (Chen et al., 2019) generalizes the AdaMM framework by combining momentum-based first-moment accumulation with AMSGrad-style second-moment normalization. ZO-AdaMU (Jiang et al., 2024) relocates momentum into the simulated perturbation step in stochastic gradient approximation. R-AdaZO (Shu et al., 2025) refines the second-moment update by using the squared first moment.

Multi-Query ZO. Averaging multiple independent perturbation estimates can reduce per-step gradient variance (Duchi et al., 2015). However, Lin et al. (2025) identified the multi-query paradox. Under a fixed query budget, the variance reduction from using more queries per update may be offset by the reduced number of parameter updates. Recent work addresses this trade-off through layer-wise independent perturbations (Peng et al., 2025), gradient normalization (Dang et al., 2026), projection alignment (Lin et al., 2025), and gradient orthogonalization (Lang et al., 2026). While these works address the multi-query paradox through various strategies, they primarily operate in full-parameter space, limiting variance reduction gains.

Theory of Adaptive ZO Update. ZO-AdaMM (Chen et al., 2019) establishes convergence for an adaptive ZO optimizer with a persistent maximum second-moment state. MUZO (Peng et al., 2025) leverages multi-query gradient averaging with Adam in full-parameter space, while R-AdaZO (Shu et al., 2025) introduces the variance-aware convergence framework by exploiting the variance reduction effect of first moment estimates and refining second moment estimates accordingly. These advances reveal that adaptively coupling noisy gradient estimates to their current normalizers requires careful theoretical treatment (Taniguchi et al., 2024).

3 PRELIMINARIES

We first introduce the foundational SubZero framework that SubZero+ builds upon. We use bold lowercase letters $\boldsymbol{w}$ to denote vectors, bold uppercase letters $\boldsymbol{W}$ to denote matrices, and calligraphic letters $\mathcal{W}$ to denote sets of matrices. $\mathrm{vec}(\boldsymbol{W})$ vectorizes $\boldsymbol{W}$ by stacking its columns into a single column vector, $\boldsymbol{A} \odot \boldsymbol{B}$ denotes element-wise multiplication, $\boldsymbol{A} \otimes \boldsymbol{B}$ denotes the Kronecker product, $\mathbb{E}[\boldsymbol{x}]$ and $\mathrm{Var}[\boldsymbol{x}]$ denote the expectation and variance of the random variable $\boldsymbol{x}$, $\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^{n} \boldsymbol{x}_i^2}$ denotes the ℓ_2-norm, and $\mathrm{bdiag}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_\ell)$ denotes a block diagonal matrix with diagonal blocks $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_\ell$.

Problem Setup. Consider fine-tuning a pre-trained LLM with l layers. Let the trainable parameters of layer i be represented by a matrix $\boldsymbol{W}_i \in \mathbb{R}^{m_i \times n_i}$ (for example, the weight matrices of linear projections in attention and feed-forward blocks). Let $\mathcal{W} = \{\boldsymbol{W}_i\}_{i=1}^{l}$ denote the set of all trainable parameter matrices, and define the flattened parameter vector $\boldsymbol{w} = \left(\mathrm{vec}(\boldsymbol{W}_1)^\top, \ldots, \mathrm{vec}(\boldsymbol{W}_l)^\top\right)^\top \in \mathbb{R}^d$, where $d = \sum_{i=1}^{l} m_i n_i$ is the total number of represented matrix parameters. The fine-tuning objective is to minimize a loss function $\mathcal{L}(\mathcal{W})$ over a training dataset $\mathcal{D}$.

SubZero. SubZero (Yu et al., 2025) reduces the variance of gradient estimation by restricting perturbations to a low-rank subspace constructed separately for each layer. For layer i, SubZero generates two column-orthonormal projection matrices $\boldsymbol{U}_i \in \mathbb{R}^{m_i \times r}$ and $\boldsymbol{V}_i \in \mathbb{R}^{n_i \times r}$ by taking the QR decomposition of random Gaussian matrices, where $r \ll \min\{m_i, n_i\}$ is the subspace rank. The layer-wise perturbation is

$$\tilde{\boldsymbol{Z}}_i = \boldsymbol{U}_i \boldsymbol{Z}_i \boldsymbol{V}_i^\top \in \mathbb{R}^{m_i \times n_i}, \tag{1}$$

where $\boldsymbol{Z}_i \in \mathbb{R}^{r \times r}$ is a low-dimensional random matrix with i.i.d. $\mathcal{N}(0,1)$ entries. The loss difference is computed globally as

$$\rho = \frac{\mathcal{L}(\mathcal{W} + \varepsilon \tilde{\mathcal{Z}}; \mathcal{B}) - \mathcal{L}(\mathcal{W} - \varepsilon \tilde{\mathcal{Z}}; \mathcal{B})}{2\varepsilon}, \tag{2}$$

where $\tilde{\mathcal{Z}} = \{\tilde{\boldsymbol{Z}}_i\}_{i=1}^{l}$ denotes the set of layer-wise perturbations. The gradient estimate for layer i is $\widehat{\nabla}_{\boldsymbol{W}_i} \mathcal{L} = \rho\, \tilde{\boldsymbol{Z}}_i$, and parameters are updated using SGD via $\boldsymbol{W}_i \leftarrow \boldsymbol{W}_i - \eta\, \widehat{\nabla}_{\boldsymbol{W}_i} \mathcal{L}$, where η denotes the learning rate.

4 THE PROPOSED SUBZERO+

SubZero+ introduces two design choices to make adaptive ZO optimization practical in random subspaces. First, it employs multi-query forward differences to reduce direction-sampling noise in the coefficient estimate. Second, it runs Adam entirely in the resulting low-dimensional coefficient matrices, so its first- and second-moment buffers remain negligible relative to the model parameters.

4.1 MULTI-QUERY ESTIMATION WITH FORWARD-DIFFERENCE

SubZero estimates the gradient from a single perturbation. To improve the coefficient signal used by Adam, SubZero+ averages multiple independent perturbation queries. Theorem 1 separates the practical one-sided estimator's finite-difference bias from its query variance, while Theorem 2 shows that averaging reduces additive second-moment inflation by $1/K$ in the exact-directional limit. Rather than computing full-parameter gradient estimates explicitly, we accumulate multi-query estimates directly within the low-dimensional coefficient space. While more queries reduce per-update direction-sampling variance, they also reduce the total number of updates available under a fixed forward-evaluation budget. The optimal query count is therefore problem-dependent and is explored in Subsection 6.3. We employ forward differences rather than central differences, requiring K perturbed evaluations plus one shared baseline evaluation per update.

Formally, let $\boldsymbol{U}_i \in \mathbb{R}^{m_i \times r}$ and $\boldsymbol{V}_i \in \mathbb{R}^{n_i \times r}$ be fixed projection matrices for layer i at the current step. They are two column-orthonormal projection matrices generated by QR decomposition of random Gaussian matrices. For query $k = 1, \ldots, K$, we independently sample $\boldsymbol{Z}_i^{(k)} \in \mathbb{R}^{r \times r}$ with i.i.d. $\mathcal{N}(0,1)$ entries and construct the full-parameter perturbation

$$\tilde{\boldsymbol{Z}}_i^{(k)} = \boldsymbol{U}_i \boldsymbol{Z}_i^{(k)} \boldsymbol{V}_i^\top. \tag{3}$$

Let $\tilde{\mathcal{Z}}^{(k)} = \{\tilde{\boldsymbol{Z}}_i^{(k)}\}_{i=1}^{l}$ denote the collection of perturbations across all l layers. We compute the baseline loss $\ell_0 = \mathcal{L}(\mathcal{W}; \mathcal{B})$ and the perturbed loss for each query

$$\ell_k = \mathcal{L}(\mathcal{W} + \varepsilon \tilde{\mathcal{Z}}^{(k)}; \mathcal{B}). \tag{4}$$

Employing a forward-difference scheme, the multi-query gradient estimate for layer i becomes

$$\widehat{\nabla}_{\boldsymbol{W}_i} \mathcal{L}(\mathcal{W}; \mathcal{B}) = \frac{1}{K} \sum_{k=1}^{K} \frac{\ell_k - \ell_0}{\varepsilon} \tilde{\boldsymbol{Z}}_i^{(k)} = \boldsymbol{U}_i \left(\frac{1}{K} \sum_{k=1}^{K} \frac{\ell_k - \ell_0}{\varepsilon} \boldsymbol{Z}_i^{(k)} \right) \boldsymbol{V}_i^\top = \boldsymbol{U}_i \overline{\boldsymbol{Z}}_i \boldsymbol{V}_i^\top. \tag{5}$$

4.2 LOW-RANK SUBSPACE ADAM

SubZero applies SGD to all coefficient directions with a common learning rate. In contrast, coefficient scales can differ substantially across layers and coordinates. SubZero+ therefore runs Adam entirely in the $r \times r$ coefficient space. For layer i at step t, let $\overline{\boldsymbol{Z}}_i^t$ denote the estimate in Eq. (5). We maintain two $r \times r$ buffers

$$\boldsymbol{M}_i^t = \beta_1 \boldsymbol{M}_i^{t-1} + (1 - \beta_1) \overline{\boldsymbol{Z}}_i^t, \tag{6}$$

$$\boldsymbol{S}_i^t = \beta_2 \boldsymbol{S}_i^{t-1} + (1 - \beta_2)(\overline{\boldsymbol{Z}}_i^t \odot \overline{\boldsymbol{Z}}_i^t), \tag{7}$$

where $0 \le \beta_1, \beta_2 < 1$, and $\epsilon_{\text{adam}} > 0$. Since the projection bases $\boldsymbol{U}_i^t$ and $\boldsymbol{V}_i^t$ remain fixed throughout each F-step block, the moment buffers preserve a consistent coordinate system across iterations within the block. To correct for EMA initialization at block entrances, we apply local bias correction using the within- block counter $c_t = 1 + ((t-1) \bmod F)$ rather than the global iteration count. The locally bias-corrected states are

$$\widehat{\boldsymbol{M}}_i^t = \boldsymbol{M}_i^t / (1 - \beta_1^{c_t}), \qquad \widehat{\boldsymbol{S}}_i^t = \boldsymbol{S}_i^t / (1 - \beta_2^{c_t}). \tag{8}$$

The ambient update is

$$\boldsymbol{W}_i^{t+1} = \boldsymbol{W}_i^t - \eta \boldsymbol{U}_i^t \left(\widehat{\boldsymbol{M}}_i^t \Big/ \left(\sqrt{\widehat{\boldsymbol{S}}_i^t} + \epsilon_{\text{adam}} \right) \right) (\boldsymbol{V}_i^t)^\top. \tag{9}$$

Algorithm 1 SignCorrectedQR(r, m_i, n_i)

Input: Rank r, dimensions m_i, n_i.

Draw $\boldsymbol{\Omega}_U \in \mathbb{R}^{m_i \times r}$, $\boldsymbol{\Omega}_V \in \mathbb{R}^{n_i \times r}$ with i.i.d. $\mathcal{N}(0,1)$
$\tilde{\boldsymbol{U}}_i, \boldsymbol{R}_U \leftarrow \mathrm{QR}(\boldsymbol{\Omega}_U)$; $\tilde{\boldsymbol{V}}_i, \boldsymbol{R}_V \leftarrow \mathrm{QR}(\boldsymbol{\Omega}_V)$
$\boldsymbol{s}_U \leftarrow (\mathrm{sgn}(\boldsymbol{R}_{U,11}), \ldots, \mathrm{sgn}(\boldsymbol{R}_{U,rr}))^\top$; $\boldsymbol{s}_V \leftarrow (\mathrm{sgn}(\boldsymbol{R}_{V,11}), \ldots, \mathrm{sgn}(\boldsymbol{R}_{V,rr}))^\top$
$\boldsymbol{U}_i \leftarrow \tilde{\boldsymbol{U}}_i \,\mathrm{diag}(\boldsymbol{s}_U)$; $\boldsymbol{V}_i \leftarrow \tilde{\boldsymbol{V}}_i \,\mathrm{diag}(\boldsymbol{s}_V)$
return $\boldsymbol{U}_i, \boldsymbol{V}_i$

Algorithm 2 PerturbParams$(\mathcal{W}, \mathcal{U}, \mathcal{V}, r, \varepsilon, s)$

Input: Model parameters $\mathcal{W}$, projection sets $\mathcal{U}$, $\mathcal{V}$, rank r, perturbation scale ε, seed s.

Reset RNG with seed s
$\mathcal{Z} \leftarrow \emptyset$
for $i = 1, 2, \ldots, l$ **do**
  Generate $\boldsymbol{Z}_i \in \mathbb{R}^{r \times r}$ with i.i.d. $\mathcal{N}(0,1)$ entries
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{\boldsymbol{Z}_i\}$
  $\boldsymbol{W}_i \leftarrow \boldsymbol{W}_i + \varepsilon \boldsymbol{U}_i \boldsymbol{Z}_i \boldsymbol{V}_i^\top$
return $\mathcal{W}, \mathcal{Z}$

Algorithm 3 SubZero+

Input: Parameter matrices $\boldsymbol{W}_i \in \mathbb{R}^{m_i \times n_i}$ for $i = 1, \ldots, l$, loss $\mathcal{L}$, step budget T, perturbation scale ε, learning rate η, subspace change frequency F, rank r, perturbation-query count K, Adam parameters $\beta_1, \beta_2, \epsilon_{\mathrm{adam}}$.

for $t = 1, 2, \ldots, T$ **do**
  Sample a minibatch $\mathcal{B}^t \subset \mathcal{D}$
  for $i = 1, 2, \ldots, l$ **do**
    if $(t-1) \bmod F \equiv 0$ **then**
      $\boldsymbol{U}_i^t, \boldsymbol{V}_i^t \leftarrow \mathrm{SignCorrectedQR}(r, m_i, n_i)$
      $\boldsymbol{M}_i^{t-1} \leftarrow \boldsymbol{0}$; $\boldsymbol{S}_i^{t-1} \leftarrow \boldsymbol{0}$ {restart Adam states before recursion}
      $c_t \leftarrow 1$ {restart bias-correction counter}
    else
      $\boldsymbol{U}_i^t \leftarrow \boldsymbol{U}_i^{t-1}$; $\boldsymbol{V}_i^t \leftarrow \boldsymbol{V}_i^{t-1}$
      $c_t \leftarrow c_{t-1} + 1$
  Compute baseline loss $\ell_0^t \leftarrow \mathcal{L}(\mathcal{W}^t; \mathcal{B}^t)$
  Initialize coefficient gradients $\overline{\boldsymbol{Z}}_i^t \leftarrow \boldsymbol{0} \in \mathbb{R}^{r \times r}$ for all i
  for $k = 1, 2, \ldots, K$ **do**
    Sample a random seed s_k^t
    $\mathcal{W}^t, \mathcal{Z}_k^t \leftarrow \mathrm{PerturbParams}(\mathcal{W}^t, \mathcal{U}^t, \mathcal{V}^t, r, \varepsilon, s_k^t)$
    $\ell_k^t \leftarrow \mathcal{L}(\mathcal{W}^t; \mathcal{B}^t)$
    $\rho_k^t \leftarrow (\ell_k^t - \ell_0^t)/\varepsilon$
    $\mathcal{W}^t, _ \leftarrow \mathrm{PerturbParams}(\mathcal{W}^t, \mathcal{U}^t, \mathcal{V}^t, r, -\varepsilon, s_k^t)$
    for $i = 1, 2, \ldots, l$ **do**
      $\overline{\boldsymbol{Z}}_i^t \leftarrow \overline{\boldsymbol{Z}}_i^t + \rho_k^t \, \boldsymbol{Z}_{k,i}^t / K$
  for $i = 1, 2, \ldots, l$ **do**
    Compute $\boldsymbol{M}_i^t$ and $\boldsymbol{S}_i^t$ by Eqs. (6) and (7)
    Bias-correct the two moments with local counter c_t by Eq. (8)
    Compute $\boldsymbol{W}_i^{t+1}$ by Eq. (9)
return $\mathcal{W}^{T+1}$

All square roots and divisions in this update are element-wise. The projection bases are periodically refreshed every $F > 1$ steps as in SubZero. The moment buffers cannot be transferred across basis refreshes because successive random subspaces are nearly orthogonal in low-rank settings, rendering previous coordinate statistics incompatible. Therefore, before the recursion at a refresh step, we set the pre recursion states $\boldsymbol{M}_i^{t-1}$ and $\boldsymbol{S}_i^{t-1}$ to zero and restart c_t. While first-moment transport across basis refreshes is theoretically possible, preliminary experiments show negligible performance gains while complicating implementation. We therefore adopt a simpler reset strategy and leave transport mechanisms for future work. Each matrix layer contributes only $2r^2$ moment floats. Separately handled one-dimensional parameters have their own scalar Adam states.

4.3 Final Algorithm

Algorithm 1 constructs the sign-corrected projection matrices following the standard QR-based approach to Haar sampling (Mezzadri, 2007). Note that standard QR implementations (e.g., `torch.linalg.qr`) do not enforce a positive diagonal in the triangular factor. Therefore, we

apply a sign correction for unique decomposition. Algorithm 2 generates and applies the perturbations, and Algorithm 3 presents the complete SubZero+ training loop. Let K be the number of perturbation queries per update: each iteration evaluates K perturbed losses plus one additional baseline loss, for a total of $K+1$ forward passes. SubZero+ remains entirely BP-free and inherits SubZero's compatibility with full-parameter tuning (FT), LoRA (Hu et al., 2022), prefix tuning (Li & Liang, 2021), and prompt tuning (Lester et al., 2021). For LoRA, following SubZero, we reshape each trainable two-dimensional factor into an approximately square matrix with the same number of elements before constructing the random subspace, and reshape each perturbation and update back to its original shape.

5 Theoretical Analysis

We analyze how multi-query averaging affects both the coefficient estimate and its squared value written to Adam. The results are conditional, single-step statements for a fixed optimization state and minibatch.

Let $q = lr^2$ and $\boldsymbol{P} = \mathrm{blkdiag}_{i=1}^{l}(\boldsymbol{V}_i \otimes \boldsymbol{U}_i) \in \mathbb{R}^{d\times q}$. For query k, stack the coefficient directions as $\boldsymbol{z}_k = (\mathrm{vec}(\boldsymbol{Z}_1^{(k)})^\top, \ldots, \mathrm{vec}(\boldsymbol{Z}_l^{(k)})^\top)^\top$. Since $\mathrm{vec}(\boldsymbol{U}_i \boldsymbol{Z}_i^{(k)} \boldsymbol{V}_i^\top) = (\boldsymbol{V}_i \otimes \boldsymbol{U}_i)\mathrm{vec}(\boldsymbol{Z}_i^{(k)})$, $\boldsymbol{P}\boldsymbol{z}_k$ is exactly the concatenated perturbation in Eq. (3), and $\boldsymbol{P}^\top\boldsymbol{P} = \boldsymbol{I}_q$. For minibatch $\mathcal{B}$, let $f_{\mathcal{B}}(\boldsymbol{w}) = \mathcal{L}(\mathcal{W};\mathcal{B})$, $\boldsymbol{a} = \boldsymbol{P}^\top \nabla_{\boldsymbol{w}} f_{\mathcal{B}}(\boldsymbol{w})$, and $a_p = \boldsymbol{e}_p^\top \boldsymbol{a}$ for $p \in [q] := \{1,\ldots,q\}$, where $\boldsymbol{e}_p$ is the pth standard basis vector.

We assume that all calls evaluate the same minibatch-loss function, with non-query evaluation randomness fixed and shared. Let $\mathcal{G}$ be the sigma-field generated by the pre-query history, $\mathcal{B}$, $\boldsymbol{w}$, $\boldsymbol{P}$, and this shared randomness; hence $f_{\mathcal{B}}$ is fixed given $\mathcal{G}$. Conditionally on $\mathcal{G}$, draw $\boldsymbol{z}_1,\ldots,\boldsymbol{z}_K \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_q)$. When $f_{\mathcal{B}}$ is differentiable at $\boldsymbol{w}$, the practical estimator and its pointwise exact-directional limit are

$$\widehat{\boldsymbol{a}}_\varepsilon = \frac{1}{K}\sum_{k=1}^{K} \frac{f_{\mathcal{B}}(\boldsymbol{w}+\varepsilon\boldsymbol{P}\boldsymbol{z}_k) - f_{\mathcal{B}}(\boldsymbol{w})}{\varepsilon}\boldsymbol{z}_k, \qquad \widehat{\boldsymbol{a}} := \lim_{\varepsilon\downarrow 0}\widehat{\boldsymbol{a}}_\varepsilon = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{z}_k\boldsymbol{z}_k^\top\boldsymbol{a}, \tag{10}$$

where the limit fixes the sampled directions, $K \in \mathbb{N}_+$, and $\varepsilon > 0$. Its ith block is $\mathrm{vec}(\overline{\boldsymbol{Z}}_i)$ from Eq. (5).

Theorem 1 (Exact-directional coefficient estimation)**.**

$$\mathbb{E}[\widehat{\boldsymbol{a}} \mid \mathcal{G}] = \boldsymbol{a}, \quad \mathrm{Cov}(\widehat{\boldsymbol{a}} \mid \mathcal{G}) = \frac{1}{K}\left(\boldsymbol{a}\boldsymbol{a}^\top + \|\boldsymbol{a}\|^2\boldsymbol{I}_q\right), \quad \mathbb{E}\left[\|\widehat{\boldsymbol{a}}-\boldsymbol{a}\|^2 \mid \mathcal{G}\right] = \frac{q+1}{K}\|\boldsymbol{a}\|^2. \tag{11}$$

Averaging K queries leaves the coefficient estimate unbiased while reducing its covariance and mean-squared error by $1/K$. For $\boldsymbol{a} \neq \boldsymbol{0}$, the relative MSE is $(q+1)/K$, depending on coefficient dimension q rather than represented dimension d.

Theorem 2 (Exact-directional squared-write inflation)**.** *For $\boldsymbol{x} \in \mathbb{R}^q$, let $\boldsymbol{x}^{\odot 2} = \boldsymbol{x}\odot\boldsymbol{x}$ denote coordinatewise squaring and let $\boldsymbol{1}_q$ be the all-ones vector. For every $p \in [q]$, define $\widehat{a}_p = \boldsymbol{e}_p^\top\widehat{\boldsymbol{a}}$. Then*

$$\mathbb{E}\left[\widehat{\boldsymbol{a}}^{\odot 2} \mid \mathcal{G}\right] = \left(1+\frac{1}{K}\right)\boldsymbol{a}^{\odot 2} + \frac{\|\boldsymbol{a}\|^2}{K}\boldsymbol{1}_q, \tag{12}$$

$$\mathbb{E}\left[\widehat{a}_p^2 \mid \mathcal{G}\right] = \left(1+\frac{1}{K}\right)a_p^2 + \frac{\|\boldsymbol{a}\|^2}{K}. \tag{13}$$

The global-energy term $\|\boldsymbol{a}\|^2/K$ means that even if $a_p = 0$, coordinate p receives positive expected second-moment input when $\boldsymbol{a} \neq \boldsymbol{0}$; averaging K queries suppresses it by $1/K$.

Connection to Existing Convergence Theory. SubZero and LOZO establish convergence for related random-subspace ZO updates under standard smoothness assumptions (Yu et al., 2025; Chen et al., 2025). Rather than repeat this generic analysis, Theorems 1 and 2 isolate SubZero+'s new multi-query estimation error and its effect on adaptive moments. Complete proofs are in Appendix A.3.

Table 1: Main results on autoregressive LLMs under the FT scheme. SubZero+ uses $K = 99$ perturbation queries plus one baseline evaluation per update. All methods use a matched 40K total forward-evaluation budget.

Model	Method / Task type	SST-2	RTE	BoolQ	WIC	MultiRC	COPA	ReCoRD	SQuAD	DROP	Avg.
		—— Classification ——					— Multi-choice —		— Generation —		
OPT-1.3B	Zero-shot	53.5	53.4	45.5	57.5	45.4	75.0	70.5	27.2	11.1	48.8
	FO (Adam)	92.0	78.0	75.8	66.0	70.9	77.0	71.1	81.7	31.1	71.5
	MeZO	90.2	64.3	65.0	58.3	59.7	74.0	71.0	76.4	22.5	64.6
	SubZero	90.9	**67.5**	65.4	58.1	59.5	75.0	70.9	76.4	25.4	65.5
	LOZO	91.1	67.2	66.7	58.9	61.4	75.0	72.1	74.5	**25.5**	65.8
	TeZO	91.3	63.2	63.4	58.5	**62.0**	**77.0**	71.8	74.1	22.2	64.8
	ZO-Muon	91.7	67.1	66.2	58.8	60.9	**77.0**	72.0	76.1	24.6	66.0
	SubZero+	**92.1**	65.7	**66.9**	**59.4**	61.4	76.0	**72.3**	**77.2**	25.4	**66.3**
LLaMA3.1-8B	Zero-shot	59.8	46.6	76.3	59.6	62.3	64.9	83.7	85.0	28.5	63.0
	FO (Adam)	95.9	88.1	87.7	76.3	89.3	84.0	75.8	91.4	70.1	84.3
	MeZO	91.6	**85.9**	83.3	66.3	86.7	89.0	82.6	90.3	65.1	82.3
	SubZero	91.2	83.8	83.3	65.5	87.2	90.0	82.6	90.9	**67.8**	82.5
	LOZO	91.2	84.5	84.5	64.3	**87.9**	90.0	81.8	89.8	66.6	82.3
	TeZO	91.2	82.7	**84.7**	64.3	87.6	89.0	81.4	90.2	65.2	81.8
	ZO-Muon	93.7	84.8	84.6	66.9	87.7	90.0	**83.1**	**91.3**	67.0	83.2
	SubZero+	**95.4**	84.8	84.2	**69.6**	87.2	**92.0**	82.0	90.7	67.5	**83.7**
OPT-13B	Zero-shot	58.8	59.6	59.0	55.0	46.9	80.0	81.2	46.2	14.6	55.7
	FO (Adam)	95.1	83.4	80.6	68.0	78.9	88.0	81.8	87.0	34.7	77.5
	MeZO	91.3	67.5	67.1	59.1	61.5	87.0	81.4	83.6	30.6	69.9
	SubZero	90.8	66.4	68.3	58.3	66.8	88.0	81.2	83.6	29.2	70.3
	LOZO	92.4	67.1	68.0	57.4	62.2	**89.0**	81.0	84.2	29.6	70.1
	TeZO	92.3	72.2	68.1	62.1	**67.0**	84.0	81.7	82.7	28.2	70.9
	ZO-Muon	**94.4**	67.5	71.9	60.5	63.3	**89.0**	**81.8**	**85.7**	**31.1**	71.7
	SubZero+	93.5	**74.0**	**75.2**	**63.2**	64.7	**89.0**	81.7	85.2	30.6	**73.0**

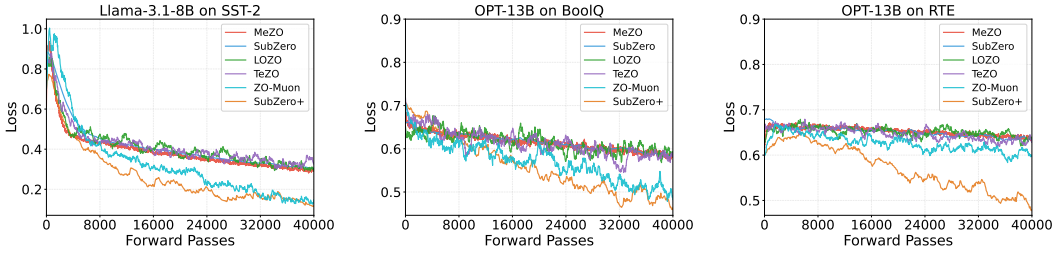


Figure 2: Selected loss trajectories corresponding to Table 1.

6 Experiments

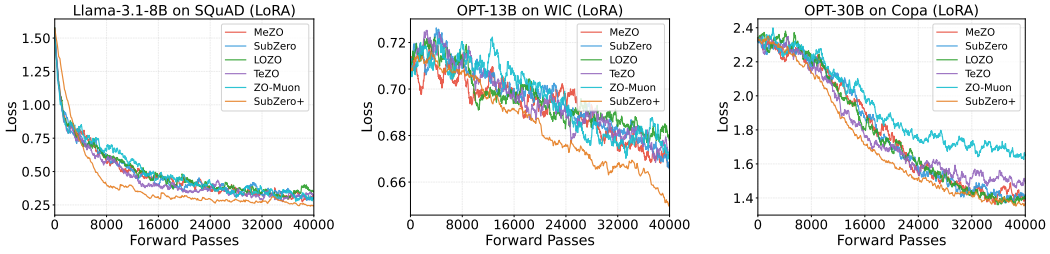
We conduct comprehensive experiments to evaluate SubZero+ against state-of-the-art ZO baselines across diverse models, tasks, and fine-tuning schemes. Due to space constraints, detailed hyperparameter configurations and additional results are provided in Appendix A.1 and A.2.

6.1 Experimental Setup

Comprehensive Validation. We evaluate autoregressive decoder-only LLMs, including OPT-1.3B, OPT-13B, OPT-30B (Zhang et al., 2022), LLaMA3.1-8B (Grattafiori et al., 2024), and Qwen2.5-32B (Team, 2024), spanning model sizes from 1.3B to 32B parameters. Tasks span the SuperGLUE benchmark (Wang et al., 2019), covering classification, multiple-choice, and generation tasks. We study both the full-parameter tuning (FT) and LoRA ($r = 8$ and $\alpha = 16$) schemes. We compare SubZero+ against MeZO (Malladi et al., 2023), SubZero (Yu et al., 2025), LOZO (Chen et al., 2025), TeZO (Sun et al., 2025), and ZO-Muon (Lang et al., 2026). For FO reference, we include Adam under FT. Zero-shot inference serves as a lower bound. MUZO (Peng et al., 2025), the most directly comparable full-space multi-query baseline, is analyzed separately due to its substantially higher memory cost from maintaining full-dimensional optimizer states. Detailed memory comparisons are given in Figure 5 of Appendix A.1.

Table 2: Main results on autoregressive LLMs under the LoRA scheme. The best result in each model/task column, including Avg., is bolded.

Model	Method	SST-2	RTE	BoolQ	WIC	MultiRC	COPA	ReCoRD	SQuAD	DROP	Avg.
	Task type	—— Classification ——					— Multi-choice —		— Generation —		
OPT-13B	MeZO	**93.6**	62.1	69.1	56.9	57.8	86.0	81.4	83.0	28.7	68.7
	SubZero	92.2	62.5	66.7	56.7	58.3	**87.0**	81.8	**83.7**	28.3	68.6
	LOZO	92.8	61.4	64.7	55.5	**60.9**	85.0	**82.1**	**83.7**	**29.5**	68.4
	TeZO	92.1	65.3	66.0	59.4	59.5	85.0	81.0	80.7	29.0	68.7
	ZO-Muon	92.9	65.3	65.9	54.9	60.0	84.0	81.3	79.4	27.6	67.9
	SubZero+	93.0	**71.5**	**73.9**	**60.2**	60.5	86.0	80.4	82.5	28.6	**70.7**
LLaMA3.1-8B	MeZO	91.3	78.0	83.6	61.0	85.8	**92.0**	83.8	90.4	65.4	81.3
	SubZero	90.6	83.8	83.5	61.1	87.3	88.0	83.5	90.4	65.3	81.5
	LOZO	89.2	78.0	83.3	64.9	86.6	87.0	82.2	90.1	66.8	80.9
	TeZO	87.2	79.8	82.1	61.1	86.8	86.0	82.6	90.8	66.3	80.3
	ZO-Muon	89.1	**86.6**	83.6	63.3	85.8	88.0	81.5	90.9	65.6	81.6
	SubZero+	**93.3**	84.8	**84.1**	**69.0**	**87.6**	89.0	**84.3**	**91.1**	**66.9**	**83.3**
OPT-30B	MeZO	92.6	68.2	63.6	60.0	65.4	**86.0**	80.5	83.6	30.6	70.1
	SubZero	93.0	65.3	61.4	58.2	64.0	85.0	**81.7**	83.2	32.5	69.4
	LOZO	93.3	66.4	64.5	60.5	60.8	85.0	81.2	82.8	32.9	69.7
	TeZO	91.5	68.2	65.5	**61.9**	61.3	85.0	80.3	81.1	30.6	69.5
	ZO-Muon	90.9	65.7	65.9	56.1	60.4	83.0	80.7	78.3	29.2	67.8
	SubZero+	**94.2**	**72.6**	**71.5**	60.3	**67.0**	**86.0**	80.9	**84.7**	**33.0**	**72.2**



(a) Llama-3.1-8B on SQuAD (b) OPT-13B on WiC (c) OPT-30B on COPA

Figure 3: Selected loss trajectories corresponding to Table 2.

Hyperparameters. SubZero+ employs default settings: perturbation queries $K = 99$ and Adam parameters $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon_{\text{adam}} = 10^{-8}$. The subspace rank r and update frequency F are given in Appendix A.2; learning rates and method-specific structural hyperparameters are tuned by grid search per benchmark. All ZO methods use batch size 16. For every method, we select the lowest-validation-loss checkpoint within each candidate run and then choose the hyperparameter configuration with the lowest validation loss (Malladi et al., 2023). All methods operate under a matched 40K forward-evaluation budget: MeZO/SubZero/LOZO/TeZO use 2 evaluations per step for 20K steps; ZO-Muon uses 5 evaluations per step for 8K steps; SubZero+ uses $K + 1$ evaluations per step with $K = 99$ and 400 steps, totaling 40,000 forward evaluations.

6.2 Main Results

6.2.1 Auto-Regressive LLMs

We first evaluate on three autoregressive models of varying scales: OPT-1.3B, LLaMA3.1-8B, and OPT-13B. Following prior ZO work (Malladi et al., 2023; Zhao et al., 2025), we use the FT scheme for all ZO methods. Results are presented in Table 1. SubZero+ attains the best average performance across all three models, and its advantage grows as model size increases. ZO-Muon also yields competitive results. Representative loss trajectories are shown in Figure 2.

We further evaluate SubZero+ under the LoRA scheme ($r = 8$, $\alpha = 16$) on OPT-13B, LLaMA3.1-8B, and OPT-30B. LoRA learns low-rank adaptation matrices for attention projections, and is widely used in practice. Results are presented in Table 2. SubZero+ achieves the best overall average on every model, with a particularly clear advantage. ZO-Muon's FT advantage fails to transfer to LoRA,

Table 3: Performance comparison on Qwen2.5-32B. Averages are computed over five tasks.

Method	SST-2	RTE	BoolQ	WIC	SQuAD	Avg.
Zero-shot	61.5	87.0	84.3	61.8	85.5	76.0
MeZO	87.5	90.3	86.7	65.7	87.9	83.6
MeZO-LoRA	62.0	87.4	84.8	64.7	84.7	76.7
SubZero	89.1	90.3	87.7	68.8	89.6	85.1
LOZO	89.4	91.0	87.0	69.3	**90.5**	85.4
ZO-Muon	90.5	**91.3**	**88.6**	71.0	**90.5**	86.4
SubZero+	**93.4**	**91.3**	88.5	**73.5**	90.3	**87.4**

Table 4: Peak GPU memory on OPT-13B (GB, batch size 16, fp16).

	FT				LoRA			
Task	MeZO	SubZero	SubZero+	Δ%	MeZO	SubZero	SubZero+	Δ%
RTE	30.4	30.5	30.7	+1.0	30.4	30.4	30.4	+0.0
BoolQ	34.1	34.2	34.5	+1.2	34.1	34.1	34.1	0.0
SQuAD	30.8	30.9	31.1	+1.0	30.8	30.8	30.8	+0.0
DROP	50.4	50.5	50.8	+0.8	50.4	50.4	50.4	0.0

likely because its matrix-sign update is designed for full-rank, near-square matrices rather than the low-rank, highly rectangular factors inherent to LoRA, whereas SubZero+ employs the reshaped subspace construction outlined in Section 4.3. Representative convergent LoRA loss trajectories for three model–task pairs are provided in Figure 3.

6.2.2 Scaling to Qwen2.5-32B

To validate SubZero+ on a substantially larger model, we additionally evaluate on Qwen2.5-32B (Team, 2024) under the FT scheme. We compare against the zero-shot baseline, MeZO, MeZO with LoRA, SubZero, LOZO, and ZO-Muon. Results are presented in Table 3. SubZero+ achieves the highest complete 5-task average.

6.3 Ablations

Memory Analysis. Table 4 shows that SubZero+ adds only $\sim 1\%$ peak-memory overhead over MeZO for OPT-13B under FT, and has zero overhead under LoRA because the trainable LoRA parameters dominate.

Learning-Rate Robustness. Figure 6 in Appendix A.1 demonstrates that SubZero+ exhibits substantially broader learning-rate tolerance than MeZO, significantly easing hyperparameter tuning in practice.

Query Count. Under a fixed 30,000-forward budget, K trades per-query variance reduction ($\propto 1/K$) against the number of steps. Table 5 in Appendix A.1 shows the optimal K is task-dependent, so improved per-update moments do not universally favor larger K.

7 Conclusion

SubZero+ enables adaptive ZO optimization through dual low-dimensionality: multi-query estimation in random subspaces and Adam updates in low-dimensional spaces. In the exact-directional limit, we show that the K-query coefficient estimator is conditionally unbiased and that its covariance, mean-squared error, and query-induced second-moment inflation decrease exactly as $1/K$. Experimentally, SubZero+ achieves consistent gains over ZO baselines across model scales from 1.3B to 32B parameters under full-parameter and LoRA schemes, with minimal memory cost.

Limitations and Future Work. Key limitations are data-independent subspace construction and momentum resets that discard temporal context. Future work should explore data-driven basis selection and moment transport across basis refreshes.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing only. We did not use generative AI tools for generating research ideas, designing the methodology, running or analyzing experiments, or producing any novel artifacts such as models, datasets, or algorithmic components. The results, claims, and experimental findings were obtained through our own work. We reviewed and verified all AI-assisted text for correctness, consistency with the paper, and absence of unsupported statements. We take responsibility for the final content of this work, including any text produced with the aid of generative AI.

Additionally, we did not use generative AI tools for any other tasks that would require disclosure. The remaining required disclosure items are therefore not applicable. We also did not use generative AI tools for any tasks that are recommended to be disclosed beyond language polishing.

ETHICS STATEMENT

We confirm that this work does not raise specific ethical concerns that require additional disclosure. It does not involve human subjects, sensitive personal data, or other practices that would introduce ethics-related risks.

SubZero+ lowers the barrier to LLM customization by reducing memory requirements and hyperparameter sensitivity. We do not identify any specific additional societal risks beyond those generally associated with LLM fine-tuning.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide relevant implementation code in the supplementary materials. The anonymous code repository is available at https://anonymous.4open.science/r/SubZero-Plus-E8A6. Readers can refer to the code and associated documentation in the supplemental materials for details needed to reproduce the experimental setup and reproduce the results. For further information, the main paper describes the methodology and experimental procedures at a level that allows verification of the reported claims, while the supplementary materials contain additional supporting details, including implementation and any relevant configuration.

REFERENCES

Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5): 185–196, 1993.

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, 2019.

Yiming Chen, Yuan Zhang, Liyuan Cao, Kun Yuan, and Zaiwen Wen. Enhancing zeroth-order fine-tuning for language models with low-rank structures. In *Proceedings of the International Conference on Learning Representations*, 2025.

Sizhe Dang, Yangyang Guo, Yanjun Zhao, Xiaodong Zheng, Guang Dai, Ivor Tsang, and Haishan Ye. FZOO: Fast zeroth-order optimizer for fine-tuning large language models towards Adam-scale speed. In *Proceedings of the International Conference on Learning Representations*, volume 2026, pp. 113760–113792, 2026.

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.

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

Athanasios Glentis, Dawei Li, Chung-Yiu Yau, and Mingyi Hong. Revisiting the Adam-SGD gap in LLM pre-training: The role of large effective learning rates. *arXiv:2605.17787*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The LLaMA 3 herd of models. *arXiv:2407.21783*, 2024.

Wentao Guo, Jikai Long, Yimeng Zeng, Zirui Liu, Xinyu Yang, Yide Ran, Jacob Gardner, Osbert Bastani, Christopher De Sa, Xiaodong Yu, Beidi Chen, and Zhaozhuo Xu. Zeroth-order fine-tuning of LLMs with transferable static sparsity. In *Proceedings of the International Conference on Learning Representations*, volume 2025, pp. 14814–14854, 2025.

Yongchang Hao, Yanshuai Cao, and Lili Mou. FLoRA: Low-rank adapters are secretly gradient compressors. In *Proceedings of the International Conference on Machine Learning*, 2024.

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 *Proceedings of the 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 *Proceedings of the International Conference on Learning Representations*, 2015.

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:2602.17155*, 2026.

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

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*, pp. 4582–4597, 2021.

Wei Lin, Qingyu Song, and Hong Xu. The multi-query paradox in zeroth-order optimization. *arXiv preprint arXiv:2509.15552*, 2025.

Sijia Liu, Yicheng Lang, Soumyadeep Pal, Changsheng Wang, Yancheng Huang, Chongyu Fan, James Diffenderfer, Bhavya Kailkhura, and Yihua Zhang. Position: Zeroth-order optimization in deep learning is underexplored, not underpowered. In *Proceedings of the International Conference on Machine Learning*, 2026.

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, 2025.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

Francesco Mezzadri. How to generate random matrices from the classical compact groups. *Notices of the American Mathematical Society*, 54(5):592–604, 2007.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 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 the Conference on Empirical Methods in Natural Language Processing*, pp. 8577–8595, 2025.

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

James C 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:2501.19057*, 2025.

Shohei Taniguchi, Keno Harada, Gouki Minegishi, Yuta Oshima, Seong Cheol Jeong, Go Nagahara, Tomoshi Iiyama, Masahiro Suzuki, Yusuke Iwasawa, and Yutaka Matsuo. ADOPT: Modified Adam can converge with any β_2 with the optimal rate. *Advances in Neural Information Processing Systems*, 37:72438–72474, 2024.

Qwen Team. Qwen2.5 technical report. *arXiv:2412.15115*, 2024.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv:1905.00537*, 2019.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *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. IEEE, 2025.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. *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 International Conference on Machine Learning*, pp. 59173–59190, 2024.

Zhong Zhang, Bang Liu, and Junming Shao. Fine-tuning happens in tiny subspaces: Exploring intrinsic task-specific subspaces of pre-trained language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 1701–1713, 2023.

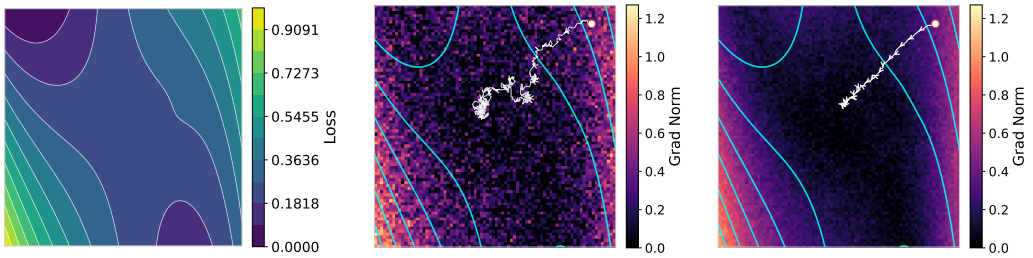
Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the International Conference on Machine Learning*, 2024.

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 *Proceedings of the International Conference on Learning Representations*, volume 2025, pp. 43496–43520, 2025.

A Appendix

A.1 Additional Experimental Results

Motivation of Multi-Query. We employ a simple 2D loss surface to demonstrate the motivation for SubZero+'s use of multi-query. The results are presented in Figure 4.



(a) Loss Landscape. (b) SubZero ($\|\overline{\boldsymbol{Z}}\|$, $K = 1$). (c) SubZero+ ($\|\overline{\boldsymbol{Z}}\|$, $K = 100$).

Figure 4: Multi-query estimation reduces gradient noise. (a) A 2D loss surface with steep walls and a flat valley. (b) SubZero with $K = 1$. The estimated gradient-norm field is noisy. (c) SubZero+ with $K = 100$. Averaging produces a visibly smoother gradient-norm field. All panels share the same domain and starting point.

Memory Cost of Full-Space Multi-Query Adam. MUZO combines multi-query averaging with Adam in the full parameter space (Peng et al., 2025). Its layer-wise seed schedule reconstructs perturbations on demand and accumulates the multi-query estimate one layer at a time. Thus, MUZO retains a current-layer gradient buffer but avoids storing all query-wise full gradients simultaneously; its persistent first- and second-moment buffers nevertheless remain $O(d)$. Accordingly, MUZO reports 35.2 GB for full-parameter MUZO-Adam versus 13.6 GB for MeZO-SGD on OPT-2.7B/SST-2, and introduces QAdam to reduce this optimizer-state cost. Our OPT-13B measurement in Figure 5 separates the same bottleneck: MeZO-style full-space Adam uses 73.77 GB even with a single query, while straightforward multi-query accumulation raises peak memory to 97.24 GB, exceeding a single 80 GB accelerator. SubZero+ instead stores the accumulated estimate and both moments in coefficient space and uses 27.93 GB.

Ablations on Query Count. Ablation study on perturbation-query count K across tasks under FT scheme. Results on OPT-1.3B with SubZero+ (sign-corrected QR subspaces, $r = 24$) across SST-2, RTE, and MultiRC. The perturbation-query budget is fixed at 30,000 for all K; larger K uses proportionally fewer steps, and each step additionally evaluates one baseline loss. All experiments use full-parameter tuning. The best accuracy for each task is bolded, and the best overall K for the average is bolded in the Avg. column.

Table 5: Ablation study on perturbation-query count K across tasks under FT scheme. Results on OPT-1.3B with SubZero+ (sign-corrected QR subspaces, $r = 24$) across SST-2, RTE, and MultiRC.

Dataset	1	5	10	20	30	50	70	80	100	150	200	Δ
SST-2	50.9	52.6	56.7	69.8	88.3	90.9	91.5	91.6	**92.2**	89.9	89.7	+41.3
RTE	47.7	52.7	49.8	57.0	62.8	**63.9**	58.1	60.3	57.0	61.4	61.0	+9.3
MultiRC	55.5	55.5	52.3	59.8	57.6	60.0	57.6	59.3	59.6	58.7	**61.1**	+4.1
Avg.	51.4	53.6	52.9	62.2	69.6	**71.6**	69.3	70.4	69.6	70.0	70.6	+18.2

Learning-Rate Robustness. We empirically compare the learning-rate (LR) robustness of the complete SubZero+ and MeZO methods by fine-tuning Qwen3-0.6B (Yang et al., 2025) with LoRA on SST-2 and sweeping learning rates across four orders of magnitude. As shown in Figure 6(a), SubZero+ achieves the highest training accuracy of 86.5% at LR $= 2 \times 10^{-2}$, and retains above 79.0% accuracy across a $3\times$ LR range ($10^{-2} \sim 3 \times 10^{-2}$). In comparison, MeZO reaches its best accuracy of 83.3% at LR $= 1 \times 10^{-4}$, but its performance window is considerably narrower. When LR is doubled to 2×10^{-4}, MeZO's accuracy drops by over 31 percentage points to 51.7%, whereas

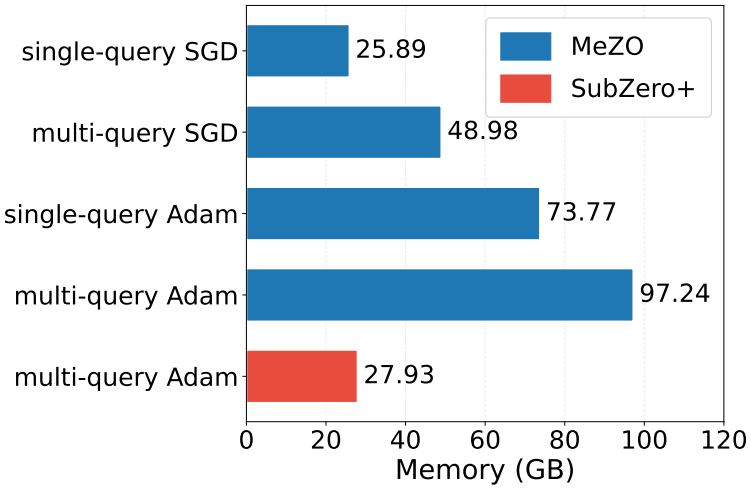
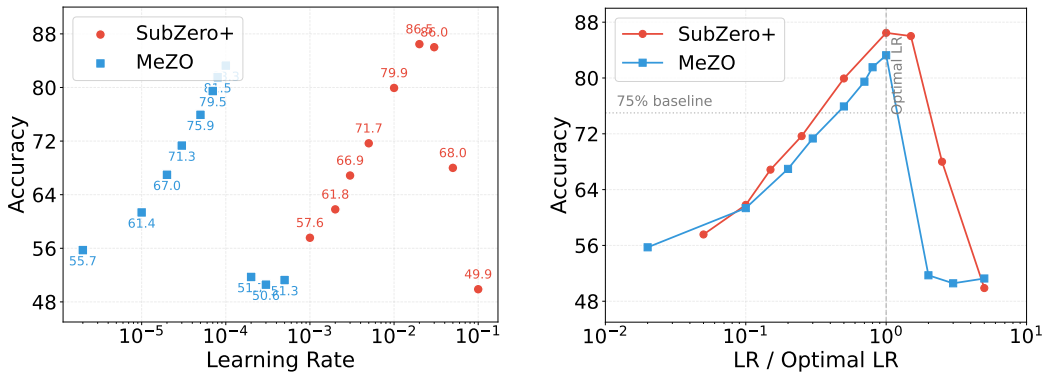


Figure 5: Peak GPU memory for full-parameter tuning of OPT-13B on SST-2 (batch size 16, fp16). The blue bars apply single- or multi-query SGD/Adam in full parameter space on top of MeZO; the red bar is SubZero+ with multi-query coefficient-space Adam. The 97.24 GB bar represents straightforward full-space query accumulation rather than a direct reproduction of MUZO's layer-wise implementation.



(a) Varying learning rates

(b) Normalized performance

Figure 6: Learning-rate robustness of SubZero+ and MeZO for LoRA of Qwen3-0.6B (Yang et al., 2025) on SST-2. (a) Training accuracy versus learning rate (LR). SubZero+ maintains high accuracy over a wide LR range, whereas MeZO degrades quickly outside its optimal value. (b) Normalized results, with each LR scaled by its optimum. SubZero+ shows a symmetric plateau, while MeZO collapses when LR is larger than the optimum.

SubZero+ drops by only 0.5 percentage points when LR is increased by $1.5\times$ to 3×10^{-2}. This comparison does not isolate the effect of multi-query averaging from the other differences between the methods.

To eliminate the scale difference in optimal LRs, we normalize each method's LR by its own optimum in Figure 6(b). The normalized view reveals that SubZero+ exhibits a broad, symmetric performance plateau around the optimal LR, with a standard deviation of merely 2.98% within the $[\text{opt}/2, 2 \times \text{opt}]$ interval. MeZO, on the other hand, shows highly asymmetric behavior: its accuracy degrades gracefully when LR is below the optimum but collapses abruptly when LR is even slightly above it. This suggests that SubZero+ is substantially more robust to learning rate selection in practical LoRA fine-tuning scenarios, where the exact optimal LR is often unknown a priori.

A.2 Experimental Details

Implementation Details. For one-dimensional parameters, SubZero uses MeZO with single-query SGD, whereas SubZero+ uses multi-query Adam. These scalar parameters are outside the Haar–Kronecker theory in Section 5. SubZero+ inherits the random-seed trick and per-layer update strategy from SubZero (Yu et al., 2025) for memory-efficient perturbation regeneration. For LoRA, also following SubZero, SubZero+ reshapes each two-dimensional trainable factor into an approximately square matrix with the same number of elements before constructing the projection bases; each perturbation and update is reshaped back to the original parameter shape before being applied. The projection matrices $\boldsymbol{U}_i, \boldsymbol{V}_i$ are regenerated every F steps and cached in between. Upon each regeneration, the moment buffers $\boldsymbol{M}_i, \boldsymbol{S}_i$ are reset to zero and their local bias-correction counter restarts. Each matrix-valued layer contributes $2r^2$ moment floats; first- and second-moment states for separately handled one-dimensional parameters are counted in addition. The cached bases require separate storage of $\sum_i (m_i + n_i)r$ floats. Multi-query perturbations are generated sequentially; each is regenerated from its seed before the forward pass and immediately subtracted afterward, keeping peak memory at inference levels.

Forward-Pass and Perturbation-Query Accounting. For the main comparison, MeZO and SubZero use two forward passes per step for 20K steps, for a total of 40K forward passes. SubZero+ uses K perturbation queries and one unperturbed baseline forward pass per step. With the default $K = 99$ and 400 steps, it uses $99 \times 400 = 39{,}600$ perturbation-query forward passes and 400 baseline forward passes, again totaling 40K forward passes. Thus, the main comparison matches total forward passes, while K always denotes the number of perturbation queries rather than the total number of forward passes per update. The query-count ablation likewise fixes a total forward-pass budget and counts the additional baseline forward pass in every update.

Hyperparameter Configurations. All experiments are conducted on a single NVIDIA H200 (141GB) GPU, except for OPT-1.3B which is run on a single NVIDIA RTX 4090 (24GB) GPU. We use PyTorch 2.1.0 with CUDA 12.1.

The method-specific hyperparameter ranges are initialized from the corresponding original papers and adapted to our model suite and common 40K-forward-pass budget. Tables 6 and 7 report the exact search grids and fixed configurations used in our experiments. Unless otherwise specified, all ZO methods use batch size 16, perturbation scale 10^{-3}, a constant learning-rate schedule, and zero weight decay. MeZO, SubZero, LOZO, and TeZO use central differences with two forward passes per optimization step and are trained for 20,000 steps. ZO-Muon uses four forward-difference perturbation queries and one shared baseline forward pass per step for 8,000 steps. SubZero+ uses 99 forward-difference perturbation queries and one shared baseline forward pass per step for 400 steps. Thus, every method uses 40,000 training-time forward passes. For every method, we select the lowest-validation-loss checkpoint within each candidate run and then choose the learning rate and method-specific structural hyperparameters by validation loss.

Table 6: Hyperparameter configurations and search grids for the FT scheme. Common settings are batch size 16, perturbation scale 10^{-3}, constant learning rate, and zero weight decay.

Method	Configuration
MeZO	learning rate $\{10^{-7}, 5\times10^{-7}, 10^{-6}\}$; 20,000 steps (2 forward passes/step)
SubZero	learning rate $\{10^{-7}, 5\times10^{-7}, 10^{-6}\}$; rank $r = 16$ (OPT-1.3B) or 128 (others); update frequency $F \in \{500, 1000, 2000\}$; 20,000 steps (2 forward passes/step)
LOZO	learning rate $\{10^{-7}, 5\times10^{-7}, 10^{-6}\}$; rank $r = 8$; reuse interval $\nu = 100$; 20,000 steps (2 forward passes/step)
TeZO	learning rate $\{10^{-7}, 5\times10^{-7}, 10^{-6}\}$; fixed rank $r = 64$; 20,000 steps (2 forward passes/step)
ZO-Muon	learning rate $\{3\times10^{-3}, 10^{-2}, 3\times10^{-2}\}$; projection rank $r \in \{64, 128\}$; $N_q = 4$; projection interval 100; 5 Newton–Schulz iterations; 8,000 steps (4 perturbation queries + 1 baseline forward pass/step)
SubZero+	learning rate $\{10^{-3}, 5\times10^{-3}, 10^{-2}\}$; rank $r = 16$ (OPT-1.3B) or 128 (others); update frequency $F \in \{50, 100\}$; $K = 99$; 400 steps (99 perturbation queries + 1 baseline forward pass/step); Adam $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\epsilon_{\text{adam}} = 10^{-8}$

Baseline Implementation Details. For LOZO (Chen et al., 2025), we use its low-rank central-difference estimator with rank $r = 8$: one Gaussian factor is reused for $\nu = 100$ consecutive steps,

Table 7: Hyperparameter configurations and search grids for the LoRA scheme. LoRA rank is 8 and $\alpha = 16$ for all methods; common ZO settings are as in Table 6.

Method	Configuration
MeZO	learning rate $\{1.5 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$; 20,000 steps (2 forward passes/step)
SubZero	learning rate $\{1.5 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$; rank $r \in \{4, 8, 16\}$; update frequency $F \in \{500, 1000, 2000\}$; 20,000 steps (2 forward passes/step)
LOZO	learning rate $\{1.5 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$; rank $r = 8$; reuse interval $\nu = 100$; 20,000 steps (2 forward passes/step)
TeZO	learning rate $\{1.5 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$; fixed rank $r = 64$; 20,000 steps (2 forward passes/step)
ZO-Muon	learning rate $\{10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$; projection rank $r \in \{8, 16\}$ (selected: 16); $N_q = 4$; projection interval 100; 5 Newton–Schulz iterations; 8,000 steps (4 perturbation queries + 1 baseline forward pass/step)
SubZero+	learning rate $\{5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}\}$; rank $r \in \{4, 8, 16\}$; update frequency $F \in \{50, 100\}$; $K = 99$; 400 steps (99 perturbation queries + 1 baseline forward pass/step); Adam $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\epsilon_{\text{adam}} = 10^{-8}$

while the complementary factor is resampled at every step. Our implementation normalizes each perturbation direction by $\sqrt{r}$; therefore, the learning rates in Tables 6 and 7 are the actual step sizes passed to our implementation rather than the α/r convention used in the original LOZO tables.

For TeZO (Sun et al., 2025), no publicly available official implementation could be located at the time of our experiments, so we implemented Algorithm 1 of the original paper. For every two-dimensional trainable parameter, the spatial factors U and V are sampled once at initialization and shared throughout training, while a temporal coefficient vector is sampled independently at every optimization step. We use the fixed-rank configuration $r = 64$; one-dimensional parameters use full-space Gaussian perturbations. Because the spatial factors remain fixed throughout training, TeZO has no periodic subspace-refresh hyperparameter in our implementation.

For ZO-Muon (Lang et al., 2026), the FT experiments follow the original choices of four forward-difference queries, projection ranks $\{64, 128\}$, projection interval 100, and Newton–Schulz matrix-sign approximation, while broadening the reported learning rate 10^{-2} to a three-point grid. The original LOZO, TeZO, and ZO-Muon papers do not report the same ZO-on-LoRA setting used in our LoRA experiments. We therefore apply their estimators to the trainable LoRA matrices while freezing the backbone. In particular, the ZO-Muon projection rank is distinct from the LoRA adapter rank; we search $\{8, 16\}$ and select 16. These LoRA results are extensions implemented in our common framework rather than direct reproductions of source-paper LoRA results.

A.3 Proofs for Section 5

All expectations below are conditional on $\mathcal{G}$. Thus $\boldsymbol{a}$, $\boldsymbol{w}$, $\boldsymbol{P}$, and $f_{\mathcal{B}}$ are fixed, while $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_K$ are independent standard Gaussian vectors.

A.3.1 Proof of Theorem 1

Proof. For one exact-directional query, define

$$\boldsymbol{x} = \boldsymbol{z}\boldsymbol{z}^\top \boldsymbol{a}. \tag{14}$$

Isserlis' identity yields

$$\mathbb{E}[\boldsymbol{x} \mid \mathcal{G}] = \boldsymbol{a}, \qquad \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\top \mid \mathcal{G}\right] = \|\boldsymbol{a}\|^2 \boldsymbol{I}_q + 2\boldsymbol{a}\boldsymbol{a}^\top. \tag{15}$$

Therefore,

$$\mathrm{Cov}(\boldsymbol{x} \mid \mathcal{G}) = \boldsymbol{a}\boldsymbol{a}^\top + \|\boldsymbol{a}\|^2 \boldsymbol{I}_q. \tag{16}$$

Averaging K conditionally independent copies preserves the mean and divides the covariance by K. Taking the trace of the resulting covariance proves the mean-squared-error identity in Eq. (11). □

A.3.2 PROOF OF THEOREM 2

Proof. The pth diagonal entry of Eq. (11) gives

$$\mathrm{Var}[\widehat{a}_p \mid \mathcal{G}] = \frac{1}{K}\left(a_p^2 + \|\boldsymbol{a}\|^2\right). \tag{17}$$

Combining this with $\mathbb{E}[\widehat{a}_p \mid \mathcal{G}] = a_p$ yields

$$\mathbb{E}\left[\widehat{a}_p^2 \mid \mathcal{G}\right] = a_p^2 + \mathrm{Var}[\widehat{a}_p \mid \mathcal{G}] = \left(1 + \frac{1}{K}\right) a_p^2 + \frac{\|\boldsymbol{a}\|^2}{K}. \tag{18}$$

Stacking the coordinate identities proves Eq. (12). □