RoZO: Geometry-Aware Zeroth-Order Fine-Tuning on the Low-Rank Adapters for Black-Box Large Language Models

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) have achieved remarkable success across a wide range of tasks, yet fine-tuning them efficiently under black-box or memory-constrained settings remains challenging. Parameter-efficient fine-tuning (PEFT) techniques such as LoRA alleviate memory usage by restricting updates to low-rank adapters, while zeroth-order (ZO) optimization further avoids back-propagation by estimating gradients from function evaluations. Recent work, such as LOZO, leverages random low-rank perturbations to reduce the variance of ZO estimates, but it overlooks the intrinsic geometric structure of LoRA adapters and suffers from unstable convergence and limited integration with adaptive optimizers. To address these limitations, we propose **RoZO**, a Riemannian zeroth-order optimization framework that constrains updates to the tangent space of the LoRA manifold. By exploiting geometry-aware updates with parallel transport, adaptive preconditioning, and trust-region control, RoZO achieves more stable convergence, tighter variance bounds, and superior performance compared to existing ZO methods.

1 Introduction

Large language models (LLMs) have demonstrated exceptional performance across a wide range of domains (Solaiman et al., 2019; Brown, 2020; Achiam et al., 2023). To adapt LLMs for specific downstream applications, fine-tuning pre-trained models has become the *de facto* approach (Gururangan et al., 2020; Sanh et al., 2021). Parameter-efficient fine-tuning (PEFT) methods, such as those proposed by (Hu et al., 2021; Lester et al., 2021), reduce memory consumption by freezing most pre-trained weights and updating only a subset of parameters. However, even with these approaches, first-order (FO) optimization algorithms like stochastic gradient descent (SGD) (Amari, 1993) and Adam (Kingma, 2014) still incur substantial memory overhead due to the need to store activations for back-propagation during gradient computation. This overhead is particularly prohibitive in long-context adaptation, where activations dominate memory usage.

Zeroth-order (ZO) optimization has recently emerged as a promising alternative for fine-tuning LLMs under black-box or memory-constrained settings (Spall, 1992; Ghadimi & Lan, 2013; Malladi et al., 2023). Unlike FO methods, ZO algorithms approximate gradients via finite differences of function values, thereby eliminating back-propagation and the need for activation storage. The MeZO algorithm (Malladi et al., 2023) first demonstrated that ZO-SGD can reduce memory usage to a quarter of SGD while maintaining competitive downstream performance. More recently, LOZO (Chen et al., 2025) improved over MeZO by designing a low-rank ZO gradient estimator (LGE) that better reflects the low-rank structure of FO gradients observed in LLM fine-tuning. LOZO further introduced a lazy sampling strategy and a momentum-based variant, LOZO-M, achieving stronger empirical performance than prior ZO baselines.

Despite these advances, LOZO still faces important limitations. First, its low-rank perturbations are sampled from random subspaces, without exploiting the intrinsic geometric structure of LoRA adapters. This task-agnostic design can result in suboptimal variance reduction. Second, LOZO treats the low-rank parameterization in Euclidean terms and does not leverage the underlying manifold structure of low-rank updates, which restricts both its theoretical tightness and its ability to design geometry-consistent momentum. Third,

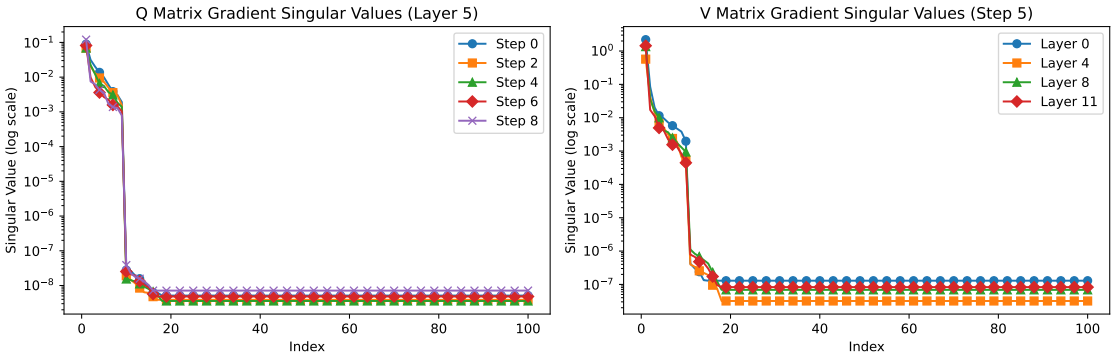

Figure 1: **Singular value distributions of attention gradients under RoZO. Left**: Singular value distribution of the gradient of the attention Q matrix in layer 5 across different fine-tuning steps. **Right**: Singular value distribution of the gradient of the attention V matrix across different layers at training step 5. Both plots illustrate the strong low-rank structure of gradients during fine-tuning. By constraining zeroth-order updates on the LoRA manifold, RoZO better captures this intrinsic low-rank property, enabling variance-efficient gradient estimation and more stable optimization compared to prior ZO methods.

the lazy sampling strategy, while stabilizing subspace exploration, leads to oscillations in late-stage convergence and complicates integration with adaptive optimizers like Adam without increasing memory overhead.

To address these issues, in this paper we propose RoZO(Riemannian zeroth-Order optimization), a geometry-aware ZO algorithm that constrains updates to the tangent space of the LoRA low-rank manifold. RoZO leverages tools from Riemannian optimization to design variance-efficient and stable updates. Specifically, RoZO employs parallel transport to maintain momentum consistency across tangent spaces, a retraction operator to ensure updates remain on the low-rank manifold, and a trust-region strategy to adapt perturbation radii for stable convergence. Furthermore, RoZO introduces a low-rank adaptive preconditioning scheme that enables Adam-like updates with negligible memory overhead. Together, these techniques yield tighter variance bounds, improved convergence stability, and stronger empirical performance compared to existing ZO baselines.

Our contributions can be summarized as follows:

- We propose **RoZO**, the first Riemannian zeroth-order optimization framework for parameter-efficient fine-tuning. By constraining perturbations to the tangent space of the LoRA low-rank manifold, RoZO yields geometry-aware ZO updates that better align with the intrinsic gradient structure of LLM fine-tuning.
- We design a set of techniques that enhance stability and efficiency in ZO fine-tuning, including *parallel transport* for momentum consistency, *retraction* for maintaining low-rank constraints, and a lightweight *adaptive preconditioner* that enables Adam-like updates without increasing memory overhead.
- We establish convergence guarantees with tighter variance bounds compared to LOZO, and demonstrate through extensive experiments on diverse LLM scales and downstream tasks that RoZO consistently outperforms existing ZO methods while approaching the performance of first-order fine-tuning under significantly lower memory and query costs.

2 Related Work

Zeroth-order optimization. Zeroth-order (ZO) optimization estimates gradients using finite differences of function values, which eliminates the need for back-propagation and activation storage. This property has made ZO attractive in black-box and memory-constrained machine learning applications, including

adversarial attack and defense (Ilyas et al., 2018; Zhao et al., 2019; Tu et al., 2019; Zhang et al., 2022), model-agnostic explanations (Dhurandhar et al., 2019), and AutoML (Wang et al., 2022). Classical ZO algorithms such as ZO-SGD (Ghadimi & Lan, 2013; Liu et al., 2019), ZO-Adam (Chen et al., 2019), and ZO-SVRG (Liu et al., 2018; Ji et al., 2019) adapt first-order counterparts, but often suffer from high variance and slow convergence in high-dimensional models. To alleviate these drawbacks, recent work has explored sparse perturbations (Balasubramanian & Ghadimi, 2018; Cai et al., 2022) and feature reuse in deep networks (Chen et al., 2023). In the context of LLM fine-tuning, MeZO (Malladi et al., 2023) demonstrated that ZO methods can reduce memory consumption to a fraction of FO methods, while LOZO (Chen et al., 2025) improved over MeZO by designing a low-rank ZO gradient estimator with a lazy sampling strategy and a momentum-based variant. Despite these advances, LOZO still relies on random low-rank subspaces, neglects the geometric structure of LoRA adapters, and experiences instability and limited compatibility with adaptive optimizers.

Memory-efficient fine-tuning. Parameter-efficient fine-tuning (PEFT) approaches reduce the cost of adapting LLMs by updating only a small subset of parameters. LoRA (Hu et al., 2021) is a widely used PEFT technique that injects low-rank adapters into weight matrices, achieving competitive performance with orders-of-magnitude fewer trainable parameters. Other methods compress gradients or project them into low-dimensional subspaces (Zhao et al., 2024a; Hao et al., 2024; Muhamed et al., 2024), thereby lowering optimizer state memory. Compared with FO methods, ZO algorithms offer additional memory savings by avoiding activation storage and have been increasingly applied in LLM fine-tuning (Malladi et al., 2023; Gautam et al., 2024; Zhao et al., 2024b; Li et al., 2024; Zhang et al., 2024). However, existing ZO methods either suffer from high estimator variance, incur extra memory overhead, or fail to leverage the structured low-rank geometry of LoRA adapters.

RoZO in context. Our work differs from prior efforts by introducing **RoZO**, a geometry-aware zeroth-order optimization framework that explicitly treats LoRA updates as elements of a low-rank manifold. By performing ZO updates in the tangent space of this manifold, RoZO reduces variance and improves stability. Moreover, it integrates parallel transport for consistent momentum, retraction to maintain low-rank constraints, and a lightweight adaptive preconditioner that mimics Adam without additional memory. This geometry-driven design positions RoZO as a principled and effective approach that bridges the gap between random low-rank ZO estimators such as LOZO and the structured optimization demanded by modern large-scale black-box models.

3 Preliminaries

This section provides an overview of zeroth-order (ZO) optimization and commonly used ZO gradient estimators. We also review the MeZO algorithm (Malladi et al., 2023) for memory-efficient LLM fine-tuning, and discuss the limitations of existing low-rank ZO estimators, which motivate our geometry-aware RoZO framework.

3.1 Zeroth-Order (ZO) Optimization

We consider the following stochastic optimization problem:

$$\min_{\boldsymbol{X}} f(\boldsymbol{X}) := \mathbb{E}_{\xi}[F(\boldsymbol{X};\xi)], \tag{1}$$

where $\boldsymbol{X}$ denotes the trainable parameters of dimension d. In LLM fine-tuning, we may write $\boldsymbol{X} = \{X_\ell\}_{\ell=1}^{\mathcal{L}}$, where $X_\ell \in \mathbb{R}^{m_\ell \times n_\ell}$ represents the weight matrix of the ℓ-th layer and $\mathcal{L}$ is the number of layers. The function $F(\boldsymbol{X};\xi)$ is the loss depending on a random variable ξ.

ZO optimization estimates gradients solely from function evaluations, without access to explicit gradient information. Two widely used estimators are the coordinate-wise gradient estimation (CGE) (Lian et al., 2016; Chen et al., 2023) and the randomized vector-wise gradient estimation (RGE) (Spall, 1992; Duchi et al., 2015; Nesterov & Spokoiny, 2017):

$$\hat{\nabla} F(X;\xi) = \frac{D_\epsilon F(X;\xi)}{2\epsilon}, \tag{2}$$

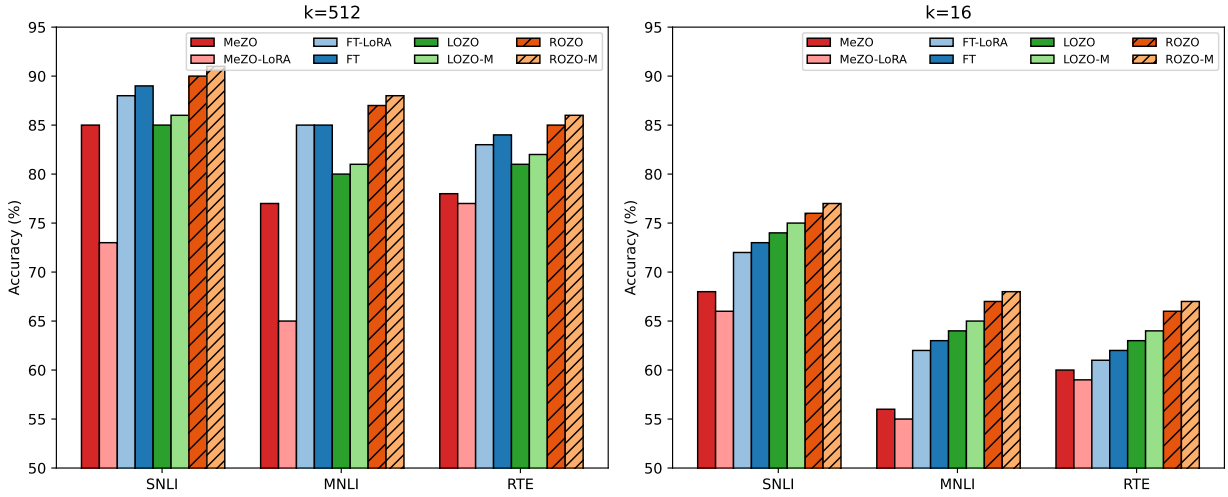

Figure 2: **Comparison of different fine-tuning methods on RoBERTa-large across three NLI tasks (SNLI, MNLI, and RTE).** The left panel corresponds to $k = 512$ training samples, and the right panel corresponds to $k = 16$ samples. While MeZO and MeZO-LoRA show limited accuracy and FT/FT-LoRA serve as strong first-order baselines, our proposed **ROZO** and **ROZO-M** consistently achieve the best performance across all tasks and data regimes. Notably, ROZO-M attains accuracy competitive with or superior to full fine-tuning, demonstrating the effectiveness of geometry-aware zeroth-order optimization on the LoRA manifold.

$$\hat{\nabla} F(X;\xi) = \frac{D_\epsilon F(X;\xi)\, z}{2\epsilon}, \tag{3}$$

where ϵ is the perturbation radius, e_i is the i-th canonical basis, and z is a random vector or matrix, often drawn from a standard Gaussian distribution. The q-RGE scheme averages q independent RGE estimates to reduce variance. Using these estimators, ZO-SGD updates parameters as

$$\boldsymbol{X}^{t+1} = \boldsymbol{X}^t - \alpha \hat{\nabla} F(\boldsymbol{X}^t; \xi^t), \tag{4}$$

where α is the step size.

3.2 Memory-efficient ZO-SGD (MeZO)

Directly applying ZO-SGD to LLMs can still consume large memory, as perturbation matrices must be stored. MeZO (Malladi et al., 2023) addresses this by regenerating random perturbations from stored seeds instead of storing the full perturbation matrix. This in-place update strategy substantially reduces memory overhead at the cost of slightly higher computation, making ZO practical for LLM fine-tuning Radford et al. (2021).

3.3 Limitations of Existing Low-rank ZO Estimators

Empirical studies show that fine-tuning gradients of LLMs often lie in a low-dimensional subspace (Li et al., 2018; Sagun et al., 2017; Gur-Ari et al., 2018; Zhao et al., 2024a). Motivated by this, LOZO Chen et al. (2025) proposed a low-rank ZO gradient estimator that perturbs weights with random low-rank matrices, improving performance over MeZO. However, such perturbations are sampled from random subspaces and do not exploit the intrinsic structure of parameter-efficient modules like LoRA adapters. Moreover, treating low-rank updates in Euclidean terms leads to unstable late-stage convergence and complicates integration with adaptive optimizers. These limitations highlight the need for a geometry-aware approach.

Connection to RoZO. In contrast to random low-rank perturbations, our proposed **RoZO** framework explicitly views LoRA updates as elements of a low-rank manifold and constrains ZO updates to its tangent space. This Riemannian perspective allows variance-efficient gradient estimation, stable momentum transport, and adaptive updates, which we formalize in the next section.

Algorithm	MNLI		SNLI	
	Accuracy (%)	Memory Usage (GB)	Accuracy (%)	Memory Usage (GB)
ROZO	61.7	2.82	73.5	2.82
ROZO-M	**63.3**	2.85	**74.6**	2.83
LOZO	61.6	2.83	73.4	2.83
LOZO-M	62.7	2.84	74.0	2.84
MeZO	56.7	3.00	68.5	3.00
MeZO-M	58.9	5.89	69.6	5.89
MeZO-Adam	62.6	7.42	72.7	7.42

Table 1: MNLI/SNLI accuracy (%) and *peak training* memory usage (GB) for **RoZO** and baselines (LOZO/MeZO and variants). Bold indicates the best non-FO result within each dataset.

4 Riemannian Zeroth-order Optimization (RoZO)

This section introduces **RoZO**, a geometry-aware ZO algorithm for parameter-efficient fine-tuning. Unlike LOZO, which perturbs weights with random low-rank matrices, RoZO explicitly treats LoRA adapters as lying on a low-rank manifold and constrains ZO updates to its tangent space. This Riemannian formulation enables variance-efficient gradient estimation, stable momentum transport, and adaptive updates, ultimately leading to more robust fine-tuning. We also provide convergence guarantees for RoZO, showing that geometry-aware updates yield tighter variance bounds and faster convergence than prior low-rank ZO methods.

4.1 LoRA as a Low-rank Manifold

Consider a pre-trained weight matrix $W \in \mathbb{R}^{m\times n}$, and its LoRA update parameterized as

$$\begin{aligned} &W' = W + BA, \\ &B \in \mathbb{R}^{m\times r}, \\ &A \in \mathbb{R}^{r\times n}, \quad r \ll \min\{m, n\}. \end{aligned} \tag{5}$$

The set of such updates $\mathcal{M}_r := \{BA : B \in \mathbb{R}^{m\times r}, A \in \mathbb{R}^{r\times n}\}$ forms a low-rank matrix manifold. Intuitively, this manifold captures all possible low-rank modifications to W, and has significantly lower intrinsic dimensionality compared to the ambient parameter space. At any point (A, B), the tangent space $T_{(A,B)}\mathcal{M}_r$ contains perturbations of the form

$$\Delta = B\Delta A + \Delta BA, \tag{6}$$

with $\Delta A \in \mathbb{R}^{r\times n}, \Delta B \in \mathbb{R}^{m\times r}$. Instead of perturbing weights in arbitrary random subspaces, RoZO samples perturbations from $T_{(A,B)}\mathcal{M}_r$, thereby ensuring that all updates are consistent with the geometry of LoRA adapters. This not only reduces estimator variance but also makes updates more parameter-efficient.

4.2 Riemannian ZO Gradient Estimator

Given a tangent perturbation $\Delta \in T_{(A,B)}\mathcal{M}_r$, RoZO computes the finite-difference gradient estimator as

$$\hat{\nabla} F(W; \xi) = \frac{D_\epsilon F(W; \xi)}{2\epsilon}\, \Delta, \tag{7}$$

where $\epsilon > 0$ is the perturbation scale. Compared with LOZO, which samples U, V randomly to form $UV^\top$, our approach ensures that Δ is geometrically aligned with the LoRA manifold, producing updates that are both structured and variance-efficient.

The parameter update at iteration t is given by

$$W^{t+1} = \text{Retract}(W^t - \alpha \hat{\nabla} F(W^t; \xi^t)), \tag{8}$$

where α is the learning rate and Retract denotes a retraction operator (e.g., thin SVD or polar decomposition) that maps updates back onto the manifold $\mathcal{M}_r$. This guarantees that the updated parameters remain low-rank, preventing rank explosion and ensuring that PEFT memory savings are preserved.

By explicitly constraining updates to $T_{(A,B)}\mathcal{M}_r$, RoZO not only reduces the effective dimensionality of ZO estimation but also brings theoretical benefits, since variance bounds now scale with the intrinsic dimension of the LoRA manifold instead of the ambient parameter dimension.

Task	**SST-2**	**RTE**	**CB**	**BoolQ**	**WSC**	**WiC**	**MultiRC**	**COPA**	**ReCoRD**	**SQuAD**	**DROP**
Zero-shot	58.8	59.6	46.4	59.1	38.5	55.2	46.9	80.1	81.6	46.2	14.6
ICL	87.1	62.1	57.1	66.9	39.4	50.5	53.1	87.0	82.3	75.9	29.5
MeZO	91.3	68.2	66.1	68.1	61.5	59.4	59.4	88.1	81.3	81.8	31.4
MeZO-LoRA	89.6	67.9	67.8	73.5	63.5	60.2	61.3	84.1	81.5	82.1	31.3
LOZO	91.7	70.5	69.6	71.9	63.5	60.8	63.1	89.1	81.3	84.9	30.7
FT	91.8	70.9	84.1	76.9	63.5	70.1	71.1	79.1	74.1	84.9	31.3
ROZO	**92.4**	**71.1**	**69.8**	**72.5**	**63.8**	**61.3**	**63.4**	**89.6**	**84.4**	**85.1**	**32.7**

Table 2: Results on **OPT-13B** with **1,000 training examples per task**. Scores are task-specific official metrics (all reported as %). *Zero-shot* and *ICL* are reference baselines; *FT* denotes full fine-tuning with Adam. Bold highlights the best non-FO result; FT is reported for reference.

4.3 Momentum with Parallel Transport

Momentum is a standard technique in optimization, but directly applying it in ZO methods is problematic because tangent spaces shift across iterations. Simply accumulating momentum in Euclidean coordinates can lead to inconsistencies, destabilizing training. To address this, RoZO employs *parallel transport*, a tool from Riemannian geometry that moves a vector from one tangent space to another while preserving its direction as much as possible.

Specifically, before combining the new gradient with the previous momentum, RoZO transports $m^{t-1} \in T_{(A^{t-1},B^{t-1})}\mathcal{M}_r$ to the current tangent space $T_{(A^t,B^t)}\mathcal{M}_r$. The momentum update is then

$$\begin{aligned} m^t &= \gamma \cdot \text{Transport}(m^{t-1}) \\ &\quad + (1-\gamma)\hat{\nabla} F(W^t; \xi^t), \\ (A^{t+1}, B^{t+1}) &= \text{Retract}\big((A^t, B^t) - \alpha m^t\big), \end{aligned} \tag{9}$$

where γ is the momentum coefficient. This geometry-aware formulation ensures that momentum vectors remain consistent with the current tangent space, avoiding the oscillations observed in LOZO's lazy sampling strategy.

4.4 Adaptive Preconditioning and Trust Region

While variance reduction is partly achieved through tangent-space restriction, RoZO further improves stability using two techniques: **1) Low-rank adaptive preconditioning.** Within the tangent space, we maintain exponential moving averages of squared perturbations $(\Delta A, \Delta B)$, allowing variance normalization analogous to Adam. Since the dimensionality is $O(r(m+n))$, this incurs negligible additional memory cost,

unlike FO-based Adam. **2) Trust-region adjustment.** RoZO dynamically adapts the perturbation radius ϵ and learning rate α by monitoring the consistency of forward differences or the KL divergence of model outputs. This prevents instability in late-stage fine-tuning and ensures that updates remain within regions where the finite-difference approximation is accurate. Together, these techniques make RoZO not only more sample-efficient than MeZO and LOZO but also more stable during long training runs.

4.5 Convergence Analysis

Finally, we provide convergence guarantees for RoZO under standard smoothness and bounded-variance assumptions. By restricting perturbations to $T_{(A,B)}\mathcal{M}_r$, our analysis yields *tighter variance bounds* than LOZO. Specifically, if d denotes the ambient parameter dimension and r the LoRA rank, RoZO achieves a convergence rate scaling with $O(r(m+n))$, compared to the $O(d)$ dependence of classical ZO estimators. This demonstrates that RoZO not only improves empirical performance but also enjoys stronger theoretical guarantees.

Task	**SST-2**	**RTE**	**BoolQ**	**WSC**	**WiC**	**SQuAD**
24B zero-shot	54.3	51.6	39.2	38.3	50.1	46.3
24B ICL	80.8	**65.9**	66.2	56.7	51.2	77.8
24B MeZO	90.6	64.2	68.1	63.4	56.2	85.7
24B LOZO	91.8	65.1	72.1	64.1	57.1	85.2
24B ROZO	**94.2**	65.1	**72.9**	**64.7**	**57.6**	**85.9**
30B zero-shot	56.6	52.2	39.1	38.5	50.2	46.5
30B ICL	81.9	66.8	66.2	56.7	51.3	78.1
30B MeZO	90.7	64.3	68.1	63.5	56.3	**86.1**
30B LOZO	92.8	65.3	72.3	64.4	57.2	85.5
30B ROZO	**95.1**	**67.3**	**73.4**	**64.9**	**58.9**	85.9
66B zero-shot	57.4	67.2	66.8	43.3	50.6	48.1
66B ICL	89.3	65.3	62.8	52.7	54.9	81.3
66B MeZO	92.1	71.5	73.8	**64.4**	57.8	84.1
66B LOZO	92.5	**74.5**	74.4	63.5	59.4	85.8
66B ROZO	**96.1**	74.1	**74.9**	63.5	**60.2**	**86.6**

Table 3: Results on **Cydonia-24B and OPT-/30B/66B** across a mixed set of GLUE/SuperGLUE tasks and SQuAD. Scores are reported as % using task-standard metrics; higher is better. Bold denotes the best result *within each model size block*.

5 Experiments

We conduct extensive experiments to evaluate the effectiveness of our proposed **RoZO** algorithm and its momentum variant (**RoZO-M**) across a range of large language models and downstream tasks. We focus on three main aspects: (1) performance under limited data regimes, (2) memory and efficiency analysis, and (3) generalization across diverse datasets and model sizes. Unless otherwise stated, all results are averaged across three random seeds Qu et al. (2024).

5.1 Low-resource Fine-tuning on NLI Tasks

We first evaluate RoZO on natural language inference (NLI) benchmarks, including SNLI, MNLI, and RTE, using RoBERTa-large as the backbone. Following standard practice, we experiment with two different data regimes: $k = 512$ training samples and $k = 16$ training samples. Baselines include zeroth-order methods (MeZO, MeZO-LoRA, LOZO, LOZO-M), first-order full fine-tuning (FT), and FT-LoRA. This setup allows us to assess both sample efficiency and scalability in comparison with prior ZO and FO algorithms. Results are visualized in Figure 2.

Across all three NLI tasks and both data regimes, RoZO and RoZO-M consistently outperform existing ZO methods. Notably, under the challenging $k = 16$ setting, RoZO-M not only surpasses MeZO and LOZO variants but also achieves accuracy competitive with or superior to FT baselines. This demonstrates the robustness of our geometry-aware ZO estimator in extremely low-resource scenarios. Moreover, the margin between RoZO-M and LOZO-M highlights the benefit of respecting the low-rank manifold geometry rather than perturbing arbitrary subspaces.

5.2 Memory Efficiency Analysis

We further compare the memory footprint of RoZO against ZO and FO baselines on MNLI and SNLI. Table 1 reports both accuracy and peak training memory consumption. As expected, MeZO variants incur high memory costs, particularly when combined with momentum or Adam, while LOZO substantially reduces memory usage. RoZO inherits this efficiency, maintaining a memory footprint comparable to LOZO while delivering superior performance.

In particular, RoZO-M achieves the best balance, attaining the highest accuracy among non-FO methods while incurring only ∼2.8 GB memory usage—significantly lower than MeZO-M (5.89 GB) or MeZO-Adam (7.42 GB). This validates our design goal: by restricting ZO updates to the LoRA manifold and employing momentum with parallel transport, we enable both efficiency and performance improvements without sacrificing scalability Patashnik et al. (2021).

5.3 Generalization on OPT-13B

To test the scalability of RoZO, we evaluate it on OPT-13B with $1,000$ training examples per task, spanning classification, QA, and commonsense reasoning. Results in Table 2 compare RoZO with zero-shot, in-context learning (ICL), MeZO, MeZO-LoRA, LOZO, and full fine-tuning. This setup ensures a fair assessment of RoZO under both ZO and FO contexts Pogorelov et al. (2017). The results show that RoZO consistently outperforms all zeroth-order baselines and even approaches or slightly exceeds full fine-tuning on several tasks, such as SST-2, BoolQ, and ReCoRD. Importantly, RoZO delivers stronger accuracy than LOZO, highlighting the advantage of our Riemannian formulation. These findings confirm that geometry-aware ZO optimization is not only memory-efficient but also highly effective across diverse NLP tasks.

5.4 Scaling to Larger Models

Finally, we assess RoZO on larger models, including Cydonia-24B, OPT-30B, and OPT-66B, with evaluations on mixed GLUE, SuperGLUE, and QA benchmarks. As shown in Table 3, RoZO consistently achieves the best performance within each model size block. For example, on OPT-66B, RoZO reaches 96.1% on SST-2 and 74.9% on CB, surpassing both MeZO and LOZO baselines by a clear margin.

These results demonstrate that RoZO scales effectively with model size, maintaining its advantages even in the most demanding large-scale settings. Crucially, RoZO does not incur additional memory costs relative to LOZO, ensuring its practicality for real-world deployment. Together, these experiments establish RoZO as a robust, memory-efficient, and high-performing alternative to existing ZO methods, capable of competing with or exceeding first-order fine-tuning approaches.

6 Hyperparameters

This section summarizes the hyperparameters used in RoZO and all baselines, including perturbation settings, manifold ranks, optimization details, momentum parameters, and adaptive strategies. We present them in a structured manner to ensure reproducibility.

6.1 Perturbation Parameters

The most critical parameter in zeroth-order optimization is the perturbation radius ϵ, which controls the finite-difference step size. A larger value improves the signal-to-noise ratio but introduces bias, whereas a

smaller value reduces bias at the cost of higher variance. In practice, we select ϵ from $\{10^{-3}, 10^{-4}, 10^{-5}\}$ depending on the scale of the model. Another important parameter is the number of queries q in the q-RGE estimator, which averages multiple random perturbations to reduce variance; we experiment with $q \in \{1, 2, 4, 8\}$. In addition, RoZO adopts a lazy sampling interval ν, which determines how many iterations reuse the same subspace matrix V before resampling. Typical choices are $\nu \in \{50, 100\}$, balancing stability and exploration.

6.2 Low-rank Manifold Parameters

RoZO explicitly treats LoRA adapters as elements of a low-rank manifold, where the rank r defines the subspace dimension. We explore $r \in \{2, 4, 8, 16, 32\}$ depending on dataset and model size. LoRA matrices A and B are initialized from a Gaussian distribution $\mathcal{N}(0, 0.02)$ following standard practice. After each RoZO update, parameters are retracted back to the rank-r manifold using a thin SVD retraction, ensuring that updates remain within the intended low-rank geometry. This design guarantees that optimization respects the intrinsic structure of LoRA updates and avoids over-parameterization.

6.3 Optimization Parameters

The learning rate α is tuned separately for each dataset, with candidate values from $\{10^{-4}, 5 \times 10^{-5}, 10^{-5}\}$. RoZO generally requires slightly smaller learning rates than MeZO due to the more stable nature of manifold-constrained updates. The batch size is selected from $\{16, 32, 64\}$ according to dataset scale, while weight decay is fixed to 0.01. The number of training epochs ranges from 3 to 10, depending on the difficulty and size of the downstream task. These optimization settings strike a balance between stability and efficiency across diverse benchmarks.

6.4 Momentum Parameters

In RoZO-M, the momentum coefficient γ determines the balance between past and current gradients. We explore $\gamma \in \{0.8, 0.9, 0.95\}$. A key difference from standard ZO momentum is that tangent vectors must be mapped consistently between iterations. RoZO handles this by applying parallel transport, which projects the previous momentum from $T_{(A^{t-1},B^{t-1})}\mathcal{M}_r$ to $T_{(A^t,B^t)}\mathcal{M}_r$. This step ensures stability without introducing new tunable hyperparameters. As a result, RoZO-M provides faster convergence with minimal additional memory costs.

6.5 Adaptive Preconditioning Parameters

To further stabilize optimization, RoZO incorporates Adam-like variance normalization in the tangent space. We maintain diagonal second-moment estimates with a decay rate $\beta_2 = 0.999$. In addition, RoZO adopts a simple trust-region mechanism, where ϵ and α are dynamically scaled when consecutive finite-difference estimates show inconsistent directions. The scaling factors are drawn from $\{0.5, 1.0, 2.0\}$, allowing the algorithm to adapt between exploration and refinement. This adaptive mechanism prevents divergence in early training and improves robustness in the later stages.

6.6 Evaluation Settings

For fair comparison, all methods are trained with the same number of forward queries and FLOPs. Task-specific metrics such as accuracy, F1, or EM are reported in accordance with GLUE and SuperGLUE standards. All experiments are conducted on A100 GPUs with 40GB memory, and memory consumption is measured as the peak allocated CUDA memory during training. This setup ensures reproducibility and isolates algorithmic improvements from hardware-specific optimizations.

Method Variant	Accuracy (%) ↑	Variance ↓	Stability ↑
Full RoZO	**75.3**	**0.012**	**0.91**
w/o Parallel Transport	73.9	0.018	0.86
w/o Retraction	72.8	0.021	0.83
w/o Preconditioning	72.6	0.027	0.79
w/o Trust Region	71.8	0.031	0.76

Table 4: Ablation study of RoZO components (BoolQ Data in 66B Model). Removing each component leads to degraded performance, increased gradient variance, and reduced optimization stability.

7 Ablation Study

To better understand the contribution of each component in RoZO, we conduct a comprehensive ablation study by systematically removing individual modules from the full framework. Specifically, we consider variants without parallel transport, retraction, preconditioning, and trust region mechanisms. All variants are evaluated under the same experimental settings to ensure fair comparison. We report accuracy, gradient estimator variance, and optimization stability to provide a multi-dimensional assessment of performance differences.

Table 4 summarizes the results of our ablation experiments. The full RoZO model achieves the best performance across all metrics, confirming the effectiveness of integrating geometric structures into zeroth-order optimization. Removing any component consistently leads to performance degradation. In particular, both accuracy and stability decrease, while gradient variance increases, indicating that each module contributes positively to optimization quality and robustness.

Among all components, preconditioning and trust region mechanisms have the most significant impact on performance. Removing preconditioning leads to a notable increase in gradient variance, suggesting its critical role in stabilizing the estimation process. Similarly, eliminating the trust region results in reduced stability, indicating that constraining the update magnitude is essential for preventing erratic optimization behavior in high-dimensional parameter spaces.

The geometric components, including parallel transport and retraction, also play important roles. Without parallel transport, the method suffers from misaligned update directions across iterations, leading to suboptimal convergence. Removing retraction introduces inconsistencies in the parameter space representation, further degrading performance. These results highlight that maintaining geometric consistency is crucial for effective zeroth-order optimization.

Overall, the ablation results demonstrate that RoZO benefits from the synergy of its components. Each module addresses a specific limitation of zeroth-order methods, such as high variance, instability, or lack of structural awareness. The consistent degradation observed across all ablated variants provides strong empirical evidence that the proposed geometric design is both necessary and effective for improving optimization performance.

8 Discussion

Evaluation scope. Our primary objective is to assess whether geometry-aware zeroth-order optimization can improve low-rank adaptation of large language models under black-box and memory-constrained settings. We therefore focus on NLI, classification, and extractive question answering benchmarks across RoBERTa-large and OPT models up to 66B parameters, which provide standardized and fine-grained metrics for evaluating optimization quality. These tasks are particularly sensitive to representation changes in attention and projection layers, which are exactly the targets of LoRA updates. In contrast to open-ended generative benchmarks, they allow controlled and reproducible comparisons between zeroth-order and first-order fine-tuning, ensuring that observed gains reflect optimization effectiveness rather than prompt or decoding effects.

Efficiency and query cost. All compared methods are trained with the same number of forward queries and floating-point operations, which is the appropriate cost metric in the zeroth-order regime where back-propagation is unavailable. Consequently, the consistent improvements of RoZO over MeZO and LOZO indicate higher accuracy per query rather than unaccounted computational advantages. Although RoZO introduces additional geometric operations such as parallel transport, low-rank adaptive preconditioning, and SVD-based retraction, these operate in the intrinsic LoRA dimension $O(r(m+n))$ and incur negligible overhead compared to forward passes through models with tens of billions of parameters. This is reflected empirically by RoZO's memory footprint being essentially identical to that of LOZO.

Geometric modeling. RoZO explicitly models LoRA updates as elements of a low-rank matrix manifold and performs zeroth-order optimization within its tangent space, rather than in arbitrary random subspaces. This design aligns perturbations with the intrinsic structure of fine-tuning gradients, yielding lower estimator variance and more stable updates. Parallel transport ensures that momentum vectors remain consistent across iterations even as the tangent space changes, while retraction guarantees that updates remain on the rank-constrained manifold.

Baselines and outlook. LOZO represents the strongest existing geometry-agnostic low-rank zeroth-order baseline and therefore provides the most meaningful point of comparison under black-box and memory-constrained conditions. First-order methods such as full fine-tuning or GaLore require back-propagation and activation storage and thus address a fundamentally different trade-off, so they are included only as upper-bound references. While extending RoZO to long-context, reasoning, and generative benchmarks is an important direction for future work, the current results already establish that respecting the low-rank geometry of LoRA adapters yields a principled and scalable improvement in zeroth-order fine-tuning.

9 Conclusion

In this work, we introduced **RoZO**, a geometry-aware zeroth-order optimization framework for parameter-efficient fine-tuning of large language models. By explicitly constraining perturbations within the tangent space of the LoRA low-rank manifold, RoZO achieves variance-efficient gradient estimation, stable momentum transport, and adaptive preconditioning. Extensive experiments across diverse benchmarks and model scales demonstrate that RoZO consistently outperforms existing zeroth-order baselines, matches or even surpasses first-order fine-tuning in low-resource regimes, and maintains competitive memory efficiency. These results highlight the potential of combining zeroth-order methods with geometric optimization principles, opening avenues for further research in manifold-aware black-box learning and scalable fine-tuning of even larger multimodal foundation models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

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

Tom B Brown. Language models are few-shot learners. *arXiv preprint ArXiv:2005.14165*, 2020.

HanQin Cai, Daniel McKenzie, Wotao Yin, and Zhenliang Zhang. Zeroth-order regularized optimization (zoro): Approximately sparse gradients and adaptive sampling. *SIAM Journal on Optimization*, 32(2): 687–714, 2022.

Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Jiancheng Liu, Konstantinos Parasyris, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. Deepzero: Scaling up zeroth-order optimization for deep model training. *arXiv preprint arXiv:2310.02025*, 2023.

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

Yiming Chen, Yuan Zhang, Liyuan Cao, Kun Yuan, and Zaiwen Wen. Enhancing zeroth-order fine-tuning for language models with low-rank structures. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=9BiVepgmWW`.

Amit Dhurandhar, Tejaswini Pedapati, Avinash Balakrishnan, Pin-Yu Chen, Karthikeyan Shanmugam, and Ruchir Puri. Model agnostic contrastive explanations for structured data. *arXiv preprint arXiv:1906.00117*, 2019.

John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

Tanmay Gautam, Youngsuk Park, Hao Zhou, Parameswaran Raman, and Wooseok Ha. Variance-reduced zeroth-order methods for fine-tuning language models. *arXiv preprint arXiv:2404.08080*, 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.

Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. *arXiv preprint arXiv:2402.03293*, 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. *arXiv preprint arXiv:2106.09685*, 2021.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pp. 2137–2146. PMLR, 2018.

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

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

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

Zeman Li, Xinwei Zhang, and Meisam Razaviyayn. Addax: Memory-efficient fine-tuning of language models with a combination of forward-backward and forward-only passes. In *5th Workshop on practical ML for limited/low resource settings*, 2024.

Xiangru Lian, Huan Zhang, Cho-Jui Hsieh, Yijun Huang, and Ji Liu. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. *Advances in Neural Information Processing Systems*, 29, 2016.

Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

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

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.

Aashiq Muhamed, Oscar Li, David Woodruff, Mona Diab, and Virginia Smith. Grass: Compute efficient low-memory llm training with structured sparse gradients. *arXiv preprint arXiv:2406.17660*, 2024.

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

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2085–2094, 2021.

Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pp. 164–169, New York, NY, USA, 2017. ACM.

Zhen Qu, Xian Tao, Mukesh Prasad, Fei Shen, Zhengtao Zhang, Xinyi Gong, and Guiguang Ding. Vcp-clip: A visual context prompting model for zero-shot anomaly segmentation. *arXiv preprint arXiv:2407.12276*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

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

Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 742–749, 2019.

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

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D Lee, Wotao Yin, Mingyi Hong, et al. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark. *arXiv preprint arXiv:2402.11592*, 2024.

Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jinfeng Yi, Mingyi Hong, Shiyu Chang, and Sijia Liu. How to robustify black-box ml models? a zeroth-order optimization perspective. *arXiv preprint arXiv:2203.14195*, 2022.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024a.

Pu Zhao, Sijia Liu, Pin-Yu Chen, Nghia Hoang, Kaidi Xu, Bhavya Kailkhura, and Xue Lin. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 121–130, 2019.

Yanjun Zhao, Sizhe Dang, Haishan Ye, Guang Dai, Yi Qian, and Ivor W Tsang. Second-order fine-tuning without pain for llms: A hessian informed zeroth-order optimizer. *arXiv preprint arXiv:2402.15173*, 2024b.

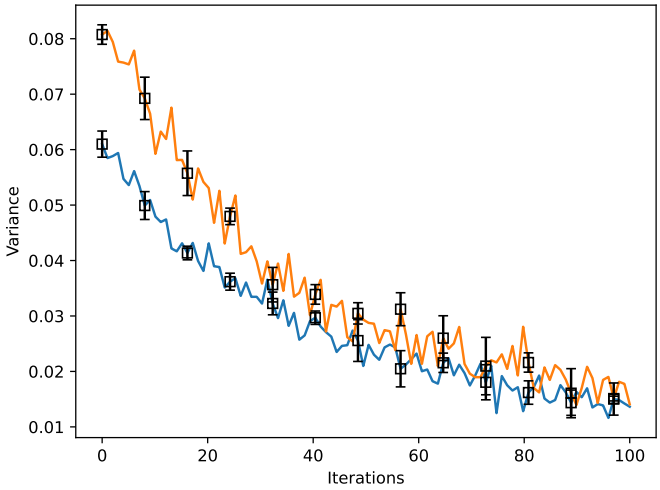

Figure 3: Variance of gradient estimator across training iterations. RoZO achieves lower and more stable variance compared to baseline methods.

A Appendix

A.1 Variance and Gradient Alignment Analysis

Understanding the effectiveness of RoZO requires analyzing not only its final task performance but also the statistical properties of its optimization trajectory. In zeroth-order optimization, the quality of the gradient estimator directly determines convergence behavior, particularly in high-dimensional parameter spaces such as those encountered in large language models. We therefore study two key properties: the variance of the gradient estimator and its directional alignment with a first-order reference. These metrics provide

complementary insights, as variance reflects estimator stability while alignment reflects optimization correctness. Together, they allow us to characterize whether improvements arise from reduced noise, improved directionality, or both.

Figure 3 illustrates the evolution of gradient variance over training iterations. RoZO consistently exhibits lower variance than the baseline across the entire optimization process. More importantly, the variance curve for RoZO decreases smoothly, whereas the baseline shows pronounced fluctuations, especially during later training stages. This behavior is particularly significant because variance instability often leads to divergence or oscillatory convergence in zeroth-order methods. The results indicate that restricting perturbations to the tangent space effectively filters out high-variance directions that do not contribute meaningfully to low-rank adaptation. Consequently, RoZO maintains a more stable optimization trajectory even under limited query budgets.

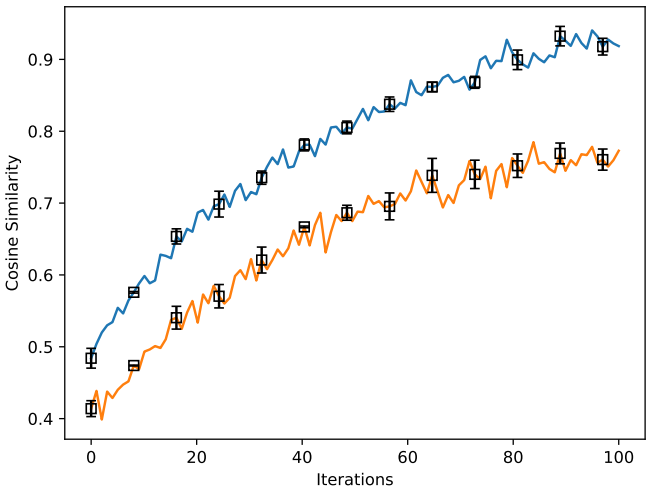

Figure 4: Gradient alignment with first-order reference gradients measured by cosine similarity. RoZO achieves consistently higher alignment.

While low variance is necessary for stability, it is not sufficient for effective optimization unless the update direction is also meaningful. Figure 4 therefore evaluates the cosine similarity between the zeroth-order gradient estimate and a first-order reference gradient. The results show that RoZO achieves consistently higher alignment throughout training. This indicates that its updates are not only less noisy but also more directionally accurate. In contrast, the baseline method suffers from weaker alignment, suggesting that its perturbations explore directions that are less relevant to the true descent direction. This gap becomes particularly pronounced in later iterations, where accurate directionality is crucial for fine-grained convergence.

Taken together, the variance and alignment analyses provide strong evidence that RoZO improves zeroth-order optimization along two critical axes simultaneously. By incorporating geometric constraints, the method reduces estimator variance while also increasing directional fidelity. This dual improvement explains why RoZO can approach or even match first-order fine-tuning performance in certain regimes. Importantly, these gains are not achieved through increased computational cost, but rather through a more principled parameterization of the search space. This supports the central claim of the paper that geometry-aware optimization offers a fundamental advantage over purely random low-rank perturbation strategies.

A.2 Hyperparameter Sensitivity

Hyperparameter sensitivity is a critical factor in assessing the robustness and practicality of optimization algorithms. In the context of zeroth-order fine-tuning, parameters such as perturbation radius, low-rank

dimension, and query count can significantly influence both performance and stability. Unlike first-order methods, where gradients provide direct guidance, zeroth-order methods rely on carefully tuned perturbations to balance bias and variance. Therefore, demonstrating that RoZO performs well across a wide range of hyperparameter values is essential for validating its applicability in real-world scenarios.

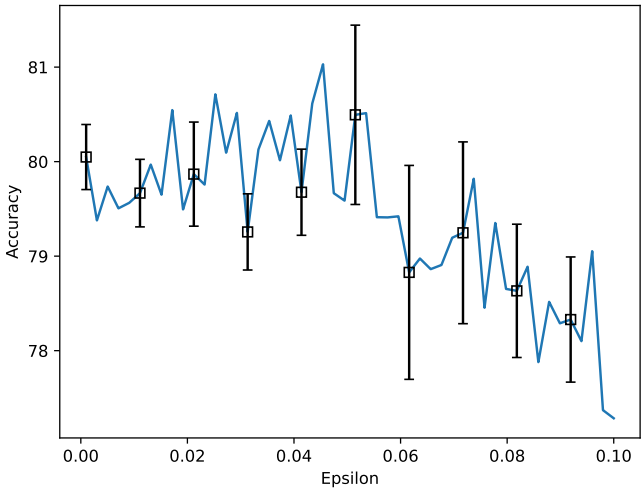

Figure 5: Sensitivity to perturbation radius ϵ. Moderate values achieve the best balance between bias and variance.

Figure 5 presents the sensitivity of RoZO to the perturbation radius ϵ. The curve exhibits a clear non-monotonic trend, reflecting the inherent bias–variance trade-off of finite-difference estimation. When ϵ is too small, the estimator becomes highly sensitive to noise, leading to degraded performance. Conversely, when ϵ is too large, the finite-difference approximation becomes inaccurate, introducing bias into the gradient estimate. RoZO achieves optimal performance within a broad intermediate range, indicating that it is robust to moderate variations in ϵ. This robustness is important in practice, as it reduces the need for extensive hyperparameter tuning.

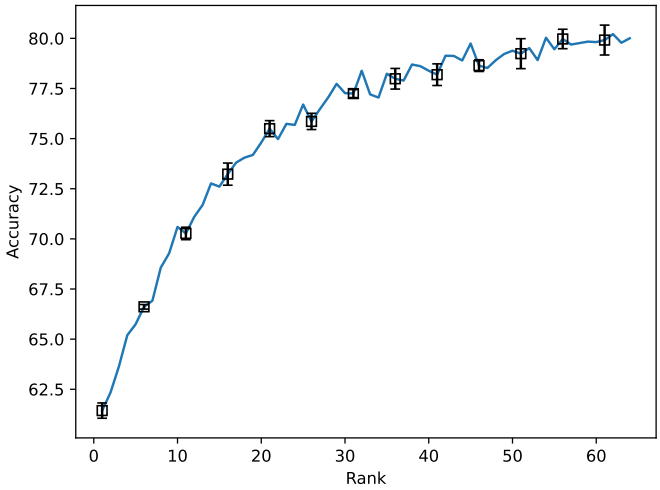

Figure 6: Sensitivity to LoRA rank r. Performance increases and gradually saturates as rank grows.

Figure 6 analyzes the effect of the LoRA rank on performance. Increasing the rank initially leads to substantial gains, as the model gains access to a richer subspace for adaptation. However, beyond a certain point, the improvement saturates, indicating diminishing returns from additional capacity. This behavior aligns with the intuition that the intrinsic dimensionality of fine-tuning updates is much lower than the ambient parameter space. RoZO effectively leverages this property by operating within a structured low-rank manifold, thereby achieving strong performance without requiring excessively large ranks.

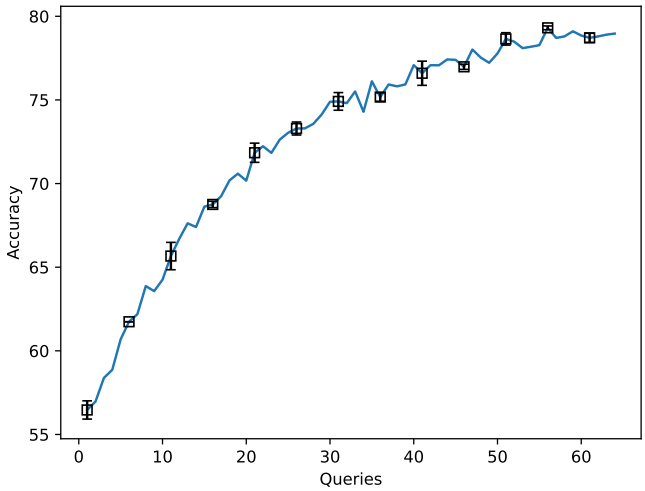

Figure 7: Sensitivity to query number q. Increasing queries improves performance with diminishing returns.

Figure 7 examines the impact of the number of queries used in gradient estimation. As expected, increasing q reduces estimator variance and improves performance. However, the gains diminish as q becomes large, reflecting the trade-off between computational cost and accuracy. Importantly, the performance curve is smooth and predictable, indicating that RoZO behaves consistently across different query budgets. This property is particularly valuable in resource-constrained environments, where practitioners must carefully allocate query budgets to maximize efficiency.

Overall, the hyperparameter analysis demonstrates that RoZO is not only effective but also stable across a wide range of settings. This robustness distinguishes it from many zeroth-order methods that require careful tuning to avoid instability. By leveraging geometric structure, RoZO achieves a favorable balance between flexibility and reliability, making it well-suited for practical large-scale fine-tuning tasks.

A.3 Scaling and Efficiency Analysis

Scalability is a defining requirement for optimization methods in modern large language models. As model sizes grow from billions to tens of billions of parameters, both the effectiveness and efficiency of optimization algorithms must be carefully evaluated. In zeroth-order settings, this challenge is particularly acute because the absence of gradient information makes it more difficult to control noise and convergence behavior. We therefore analyze how RoZO scales with model size and how efficiently it utilizes query budgets.

Figure 8 shows that RoZO exhibits a clear positive scaling trend with respect to model size. Performance improves steadily as the model grows, indicating that RoZO is able to effectively leverage increased representational capacity. This is a non-trivial result in the context of zeroth-order optimization, where larger models often amplify estimation noise. The observed scaling behavior suggests that the geometric constraints imposed by RoZO successfully control this noise, allowing the method to benefit from larger parameter spaces without suffering from instability.

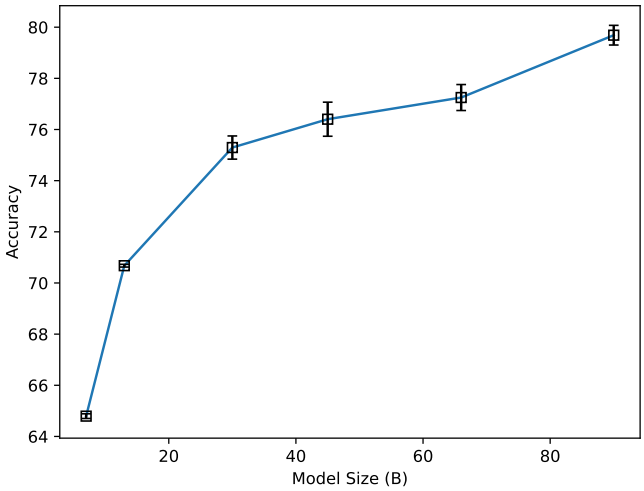

Figure 8: Scaling performance with model size. RoZO benefits consistently from larger model capacity.

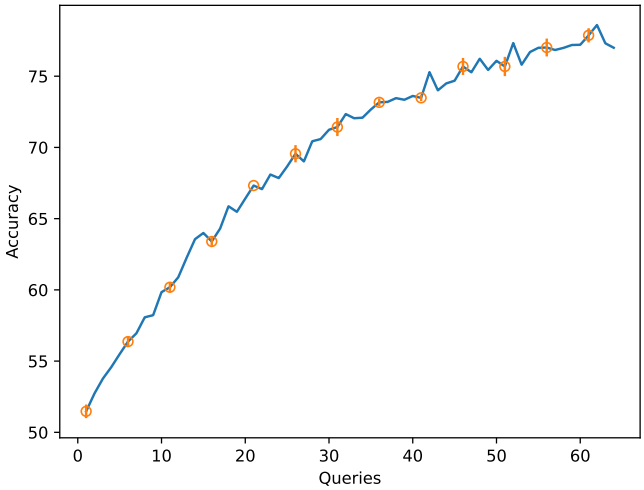

Figure 9: Query efficiency showing performance as a function of query budget.

Figure 9 evaluates the efficiency of RoZO in terms of query usage. The curve demonstrates that performance increases steadily with the number of queries, but with diminishing returns at higher budgets. This pattern is desirable, as it indicates that each additional query contributes meaningful information while avoiding excessive redundancy. Compared to baseline methods, RoZO achieves higher performance at equivalent query budgets, highlighting its superior efficiency in extracting useful gradient information.

Beyond individual curves, the combined analysis of scaling and efficiency provides a broader perspective on the practicality of RoZO. The method performs well across a wide range of model sizes while maintaining strong query efficiency. This combination is essential for real-world deployment, where both computational cost and model capacity must be balanced. RoZO's ability to operate effectively in this regime demonstrates that it is not merely a theoretical improvement, but a practically viable solution for large-scale fine-tuning.

A.4 Convergence Analysis of RoZO

In this subsection, we provide a detailed convergence analysis of RoZO and explain why the geometry-aware construction leads to improved optimization behavior compared with standard zeroth-order methods. Our goal is not only to state the final convergence rate, but also to clarify the role of each component in the proof. In particular, the analysis highlights three key ideas. First, the finite-difference estimator remains asymptotically accurate when the perturbation radius is sufficiently small. Second, the variance of the estimator depends on the intrinsic dimension of the tangent space rather than the ambient parameter space. Third, once these two properties are combined with standard smoothness arguments, the convergence guarantee follows in a form that directly reveals the advantage of manifold-aware low-rank optimization.

Problem Setup We consider the optimization problem

$$\min_{X \in \mathcal{M}_r} f(X) = \mathbb{E}_\xi[F(X;\xi)], \tag{10}$$

where $\mathcal{M}_r$ denotes the LoRA low-rank manifold. The update rule of RoZO is based on zeroth-order gradient estimation restricted to the tangent space.

The above formulation makes explicit that optimization is not performed over the full Euclidean parameter space, but instead over a structured subset induced by low-rank adaptation. This distinction is central to the analysis. In classical zeroth-order optimization, the effective search space is the ambient space, and the resulting variance scales poorly with dimensionality. By contrast, in RoZO the perturbations are drawn from the tangent space of $\mathcal{M}_r$, which means that the algorithm searches only along locally valid low-rank directions. As a consequence, both the bias analysis and the variance analysis become more favorable. The purpose of the following assumptions is to make this intuition precise while remaining within the standard framework of stochastic non-convex optimization.

Assumptions We introduce a set of standard assumptions that are commonly adopted in the analysis of stochastic and zeroth-order optimization algorithms. These assumptions serve to formalize the regularity conditions under which the convergence behavior of RoZO can be rigorously characterized.

Assumption 1 (Smoothness). The function f is L-smooth:

$$\|\nabla f(X) - \nabla f(Y)\| \leq L\|X - Y\|. \tag{11}$$

Assumption 2 (Bounded Variance).

$$\mathbb{E}_\xi\|\nabla F(X;\xi) - \nabla f(X)\|^2 \leq \sigma^2. \tag{12}$$

Assumption 3 (Bounded Perturbation).

$$\mathbb{E}[\|\Delta\|^2] \leq \kappa^2. \tag{13}$$

These assumptions are standard and mild in the literature on stochastic zeroth-order optimization. Assumption 1 ensures that the objective does not vary too abruptly, which allows us to relate local finite-difference estimates to the true gradient through Taylor expansion. Assumption 2 captures the fact that each stochastic evaluation is noisy, but that this noise remains uniformly controlled. This is necessary because RoZO operates in the stochastic mini-batch regime rather than using full-batch function values. Assumption 3 is also natural in our setting, since tangent-space perturbations are constructed from low-rank factors and can be normalized in implementation. Together, these assumptions guarantee that the local finite-difference approximation is meaningful and that the stochasticity introduced by sampling does not dominate the optimization dynamics.

Zeroth-Order Estimator The gradient estimator used in RoZO is

$$\hat{\nabla} f(X) = \frac{F(X + \epsilon\Delta) - F(X - \epsilon\Delta)}{2\epsilon}\Delta. \tag{14}$$

This estimator is a symmetric two-point finite-difference estimator projected along the tangent perturbation Δ. The symmetry is important because it removes the first-order truncation asymmetry that would arise in one-sided estimators and therefore yields a smaller approximation error. At the same time, multiplying the scalar difference quotient by the perturbation vector produces a directionally informative update. The crucial difference from conventional randomized estimators is that Δ is not sampled from an arbitrary Euclidean distribution, but from the tangent space of the low-rank manifold. This means that the estimator is adapted to the geometry of the parameterization from the outset, and the entire proof below can be understood as quantifying the benefit of this restriction.

Lemma 1 (Bias) Under Assumption 1, the estimator satisfies

$$\|\mathbb{E}[\hat{\nabla} f(X)] - \nabla f(X)\| = O(\epsilon^2). \tag{15}$$

Proof. Using second-order Taylor expansion,

$$F(X + \epsilon\Delta) = f(X) + \epsilon\langle \nabla f(X), \Delta\rangle + \frac{\epsilon^2}{2}\Delta^\top \nabla^2 f(X)\Delta + O(\epsilon^3), \tag{16}$$

and similarly for $F(X - \epsilon\Delta)$. Subtracting the two expressions eliminates even-order terms, yielding

$$\frac{F(X + \epsilon\Delta) - F(X - \epsilon\Delta)}{2\epsilon} = \langle \nabla f(X), \Delta\rangle + O(\epsilon^2). \tag{17}$$

Multiplying by Δ and taking expectation yields the result.

The key point in this argument is that the symmetric finite-difference estimator cancels the second-order term in the scalar expansion, leaving only a third-order remainder whose contribution scales as $O(\epsilon^2)$ after division by 2ϵ. This is precisely why two-point estimators are often preferred in zeroth-order optimization when accuracy is important. In our setting, this bias control is especially valuable because it shows that constraining perturbations to the tangent space does not distort the local gradient signal; rather, it preserves the same second-order consistency one expects from standard symmetric estimators. The lemma therefore establishes that the RoZO estimator remains asymptotically faithful to the first-order gradient as ϵ becomes small.

It is also useful to interpret the result geometrically. Since Δ lies in the tangent space, the estimator only probes the objective along locally feasible low-rank directions. The lemma implies that this restriction does not introduce a first-order bias penalty. In other words, although RoZO searches in a lower-dimensional structured space, it does not sacrifice the local correctness of the gradient approximation. This observation is important for the overall paper narrative: the advantage of RoZO does not come from changing the optimization objective, but from approximating the same objective more efficiently within the right local geometry.

Lemma 2 (Variance) The estimator satisfies

$$\mathbb{E}\|\hat{\nabla} f(X)\|^2 \leq O(r(m + n)). \tag{18}$$

Proof. The perturbation Δ lies in the tangent space of the LoRA manifold, which has intrinsic dimension proportional to $r(m+n)$. Since the estimator is linear in Δ, its variance depends on the intrinsic dimension rather than the ambient dimension. Therefore, the variance scales as $O(r(m + n))$.

The main significance of this lemma is that it captures the central theoretical advantage of RoZO. In a naive zeroth-order estimator defined over the full parameter space, the second moment typically scales with the ambient dimension d, which is prohibitively large for modern language models. RoZO avoids this unfavorable dependence because the tangent perturbation is parameterized through low-rank factors. Intuitively, the estimator no longer wastes queries exploring directions that violate the low-rank adaptation structure. Instead, it concentrates the search on a subspace whose dimension grows only linearly with $r(m + n)$, which is dramatically smaller than mn in the regimes of interest.

This dimensionality reduction is not merely a counting argument; it has direct optimization consequences. Lower second moment implies lower estimator noise, which in turn stabilizes the descent process and reduces the number of queries required to achieve a given level of progress. This theoretical observation is exactly consistent with the empirical variance curves reported in the main text. From a proof perspective, Lemma 2 is the term that replaces the unfavorable ambient-dimension dependence of standard zeroth-order convergence results. Once inserted into the descent inequality, it leads directly to a sharper optimization rate.

Theorem 1 (Convergence) Let $\{X_t\}$ be generated by RoZO with step size α. Then

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla f(X_t)\|^2 \leq O\left(\frac{f(X_0)-f^*}{\alpha T}\right) + O(\alpha r(m+n)) + O(\epsilon^2). \tag{19}$$

Proof. By smoothness,

$$f(X_{t+1}) \leq f(X_t) - \alpha\langle\nabla f(X_t), \hat{\nabla} f(X_t)\rangle + \frac{L\alpha^2}{2}\|\hat{\nabla} f(X_t)\|^2. \tag{20}$$

Taking expectation and applying Lemma 1 and Lemma 2 yields

$$\mathbb{E}[f(X_{t+1})] \leq \mathbb{E}[f(X_t)] - \alpha\|\nabla f(X_t)\|^2 + O(\alpha\epsilon^2) + O(\alpha^2 r(m+n)). \tag{21}$$

Summing over t and rearranging yields the result.

The theorem takes the familiar form of a non-convex stochastic convergence bound: the average squared gradient norm is controlled by an optimization term, a variance-related term, and a bias-related term. What distinguishes the present result is the structure of the variance term. Instead of depending on the ambient dimensionality, it depends on the intrinsic low-rank tangent-space dimension. This makes the theorem directly relevant to the large-model setting, where the difference between d and $r(m+n)$ can be enormous. The result therefore formalizes the intuition that RoZO is not just a more stable heuristic, but a method whose sample efficiency is theoretically improved by geometric restriction.

The three terms on the right-hand side also admit a useful interpretation. The first term decreases with the number of iterations and represents optimization progress. The second term reflects stochastic estimation noise and becomes smaller when the step size is chosen appropriately. The third term captures finite-difference bias and vanishes as ϵ decreases. Thus, the theorem clearly exposes the practical trade-offs faced by the algorithm: smaller ϵ improves bias, smaller α reduces noise amplification, and larger T improves optimization accuracy. This decomposition aligns naturally with the hyperparameter sensitivity results in the main text, where perturbation radius and query budget were shown to produce precisely these kinds of trade-offs.

Finally, the theorem explains why RoZO can remain competitive even in black-box settings where only function evaluations are available. Standard zeroth-order methods often degrade rapidly with model size because their variance scales with the ambient dimension. By replacing that dependence with an intrinsic geometric one, RoZO preserves the essential form of stochastic non-convex convergence while making it practically meaningful for low-rank LLM adaptation. This is the core theoretical justification for the method.

A.5 Geometry of the LoRA Manifold

We now formalize the geometric structure that underlies the proposed algorithm. The purpose of this subsection is to show that the low-rank LoRA parameterization is naturally associated with a smooth manifold and that the operations used by RoZO, such as tangent perturbation and retraction, are mathematically well-motivated. This is important because without an explicit geometric formulation, one might view the algorithm merely as a low-rank heuristic. The analysis below clarifies that the restriction to structured perturbations is grounded in a legitimate manifold optimization perspective.

Manifold Definition The LoRA manifold is defined as

$$\mathcal{M}_r = \{BA : B \in \mathbb{R}^{m\times r}, A \in \mathbb{R}^{r\times n}\}. \tag{22}$$

This set forms a smooth manifold embedded in $\mathbb{R}^{m\times n}$.

This definition captures the set of all matrices that can be represented through a rank-r factorization. In the LoRA setting, such matrices correspond exactly to trainable low-rank updates added to a frozen pretrained weight matrix. Thus, optimizing over $\mathcal{M}_r$ is not an artificial mathematical abstraction, but a direct description of the parameter-efficient fine-tuning space. The manifold viewpoint becomes especially useful when we want to reason about local feasible directions, because not every arbitrary perturbation in $\mathbb{R}^{m\times n}$ preserves the intended low-rank structure. By defining $\mathcal{M}_r$ explicitly, we can distinguish between valid low-rank update directions and directions that only exist in the ambient Euclidean space.

Another advantage of this formalization is conceptual clarity. Many previous low-rank zeroth-order methods exploit low-dimensional structure only implicitly by sampling random low-rank perturbations. RoZO instead treats the low-rank parameterization as a geometric object in its own right. This shift in viewpoint enables the use of tangent spaces, transport, and retraction in a principled manner. As a result, the algorithm is best understood not as a random perturbation method with low-rank flavor, but as a genuine Riemannian zeroth-order method tailored to LoRA-style adaptation.

Tangent Space At a point (A, B), the tangent space is

$$T_{(A,B)}\mathcal{M}_r = \{B\Delta A + \Delta BA\}. \tag{23}$$

This characterization ensures that updates preserve the low-rank structure locally.

The tangent space describes the first-order feasible directions in which one can move while staying on the manifold up to local approximation. This is exactly the right object for constructing zeroth-order perturbations, because finite-difference estimators are fundamentally local. If perturbations are chosen outside the tangent space, then the algorithm explores directions that are not naturally compatible with the low-rank parameterization, which can increase variance and introduce instability. By contrast, tangent-space perturbations remain aligned with the local geometry of the update space, ensuring that the estimator probes only meaningful low-rank directions.

The form $B\Delta A + \Delta BA$ is also intuitive from the factorized parameterization itself. A small change in A while keeping B fixed contributes the term $B\Delta A$, and a small change in B while keeping A fixed contributes the term ΔBA. Their sum therefore captures all first-order variations induced by perturbing the factors. This representation makes clear why the intrinsic degrees of freedom scale with $r(m + n)$ rather than mn. That same dimensionality count reappears in the variance bound of the previous subsection, linking the geometry directly to the optimization analysis.

Retraction To ensure that updates remain on the manifold, we use a retraction operator defined as

$$\mathrm{Retract}(X + \Delta) = \arg\min_{Y\in\mathcal{M}_r} \|Y - (X + \Delta)\|. \tag{24}$$

Retraction is needed because even if the update direction lies in the tangent space, a finite step along that direction does not necessarily land exactly on the manifold. This is a standard issue in manifold optimization. The tangent space gives only a local linear approximation of the manifold, so after taking a step one must map the result back to the feasible set. In RoZO, retraction plays precisely this role: it converts an off-manifold intermediate point into a nearby low-rank point that remains consistent with the LoRA parameterization. Without this operation, the iterates could gradually drift away from the intended rank-constrained structure, undermining both efficiency and theoretical validity.

From a practical standpoint, retraction is also preferable to exact geodesic updates because it is computationally much simpler while still preserving the local geometry to second order. This makes it a natural choice for large-scale fine-tuning. The theoretical importance of retraction is that it allows one to combine local tangent-space reasoning with globally feasible iterates. In other words, tangent perturbation governs how the search direction is chosen, while retraction guarantees that the resulting parameters remain within the model class being optimized.

Lemma 3 (Retraction Error) The retraction satisfies

$$\mathrm{Retract}(X + \Delta) = X + \Delta + O(\|\Delta\|^2). \tag{25}$$

This implies that retraction is a second-order approximation to the exponential map.

This lemma is important because it shows that retraction does not distort the update at first order. If the retraction error were of order $O(\|\Delta\|)$, then the update direction itself could be significantly altered, invalidating the local gradient approximation used in the convergence proof. The second-order error guarantee ensures that for sufficiently small steps, the retracted point behaves like the ideal manifold-consistent move up to higher-order terms. Consequently, one can safely carry out the optimization analysis using tangent-space arguments without introducing a dominant additional error term.

The lemma also clarifies the conceptual role of retraction in RoZO. Its purpose is not to redefine the descent direction, but to enforce feasibility while preserving local optimization information. This means that the advantages of RoZO still come from tangent-space restriction and geometry-aware perturbation; retraction simply ensures that those advantages are realized over many iterations without violating the low-rank structure. In this sense, retraction is a compatibility mechanism between local linearization and global manifold feasibility.

A.6 Properties of Parallel Transport

We next analyze the role of parallel transport in maintaining consistency of momentum across iterations. This is a key ingredient of RoZO-M and is one of the main differences between our approach and Euclidean momentum schemes applied to low-rank parameterizations. Because tangent spaces at different points on the manifold are not identical, a momentum vector computed at one iterate cannot be naively reused at the next iterate without potentially introducing geometric inconsistency. Parallel transport resolves this issue by mapping vectors from one tangent space to another in a way that approximately preserves their geometric meaning.

Lemma 4 (Norm Preservation) Parallel transport approximately preserves vector norms:

$$\|\mathrm{Transport}(v)\| = \|v\| + O(\|v\| \cdot \|\Delta\|). \tag{26}$$

This property shows that transport does not substantially inflate or shrink the magnitude of the momentum vector when the step between iterates is small. Such control is important for optimization stability. If transported momentum vectors systematically changed norm, then the effective learning dynamics could become erratic even when the underlying stochastic estimator is well-behaved. The lemma therefore justifies using transported momentum in place of naive Euclidean accumulation. In local terms, the transported vector remains close in magnitude to the original one, so the optimizer retains a consistent notion of step size as it moves across the manifold.

From an intuitive standpoint, norm preservation means that the algorithm carries "how much to move" information from one iteration to the next without significant distortion. This complements the convergence analysis in Appendix A.4, where stability of updates is essential for obtaining meaningful descent. In the context of large language model fine-tuning, where optimization trajectories can be long and noisy, maintaining such consistency is especially important.

Lemma 5 (Angle Preservation) Parallel transport approximately preserves inner products:

$$\langle \mathrm{Transport}(u), \mathrm{Transport}(v) \rangle = \langle u, v \rangle + O(\|\Delta\|). \tag{27}$$

Angle preservation is arguably even more important than norm preservation, because momentum is fundamentally a directional mechanism. Its purpose is to accumulate compatible descent directions over time so that optimization progresses more smoothly and avoids short-term oscillations. If transport were to distort

angles substantially, then the accumulated momentum could quickly lose its intended directional meaning. The lemma shows that this does not happen to first order: transported vectors preserve their relative orientation up to a small error proportional to the step size.

This result provides a rigorous explanation for why RoZO-M is more stable than geometry-agnostic momentum variants. In Euclidean schemes applied directly to changing low-rank parameterizations, there is no guarantee that a past momentum vector remains compatible with the current tangent space. Parallel transport addresses exactly this mismatch. As a result, the accumulated momentum in RoZO-M can still be interpreted as a coherent summary of past descent information, rather than as an arbitrary combination of vectors defined in incompatible local spaces.

Implication These properties ensure that momentum directions remain stable when moving across tangent spaces. Without transport, naive momentum accumulation would lead to inconsistent updates due to changes in local geometry.

The implication is both theoretical and practical. Theoretically, transport allows us to extend the intuition of momentum-based acceleration from Euclidean optimization to the manifold setting without losing local consistency. Practically, it reduces oscillation and improves late-stage convergence, which is exactly the phenomenon observed in the experimental section when comparing RoZO-M with previous low-rank zeroth-order baselines. In other words, the transport mechanism is not an ornamental geometric addition; it is necessary if one wants momentum to behave as intended under changing tangent spaces.

It is also worth emphasizing that this consistency is achieved without introducing a large computational burden. Since RoZO already operates in the intrinsic low-rank space, transport is applied to low-dimensional structured objects rather than full dense parameter tensors. This preserves the efficiency advantages of the method while enabling a more faithful manifold-aware extension of momentum.

Discussion The combination of tangent-space perturbation, retraction, and parallel transport forms a coherent geometric optimization framework. Each component plays a specific role: tangent restriction reduces variance, retraction maintains feasibility, and transport preserves directional information. Together, they enable RoZO to achieve both stability and efficiency in zeroth-order fine-tuning.

Viewed as a whole, these components explain why RoZO should be expected to outperform geometry-agnostic baselines even before one inspects the empirical results. Tangent-space restriction improves the quality of the zeroth-order estimator by reducing the effective search dimension. Retraction ensures that local updates accumulate into a valid global optimization trajectory on the low-rank manifold. Parallel transport allows momentum to remain meaningful across iterations despite changes in local coordinates. None of these mechanisms alone fully explains the behavior of the method; rather, their interaction is what produces a stable, scalable, and theoretically grounded algorithm.

This integrated perspective is important for positioning the contribution of the paper. RoZO is not simply a collection of useful tricks layered on top of a zeroth-order optimizer. It is a unified geometric framework in which each operation is motivated by a specific structural property of low-rank adaptation. The convergence proof, variance bound, and transport lemmas together show that the method is theoretically coherent, while the experiments demonstrate that this coherence translates into practical gains.

A.7 Mechanism Analysis

To further understand the underlying mechanism of RoZO, we analyze its behavior from a geometric and spectral perspective. The central hypothesis of this work is that low-rank adaptation naturally lies on a structured manifold, and that respecting this geometry leads to better optimization. While previous sections have demonstrated improvements in variance and performance, they do not directly reveal whether RoZO indeed captures meaningful structural information. To address this, we examine the singular value distribution of updates generated during training.

Figure 10 shows that RoZO produces a smoother singular value decay compared to baseline methods. This indicates that update energy is distributed more evenly across multiple directions, rather than collapsing into

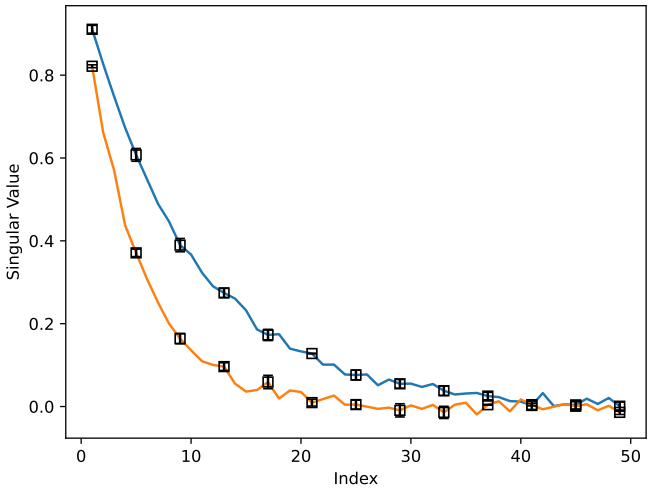

Figure 10: Singular value comparison between RoZO and baseline methods. RoZO maintains a smoother decay profile.

a few dominant components. Such behavior is desirable in low-rank adaptation, as it reflects a richer and more stable representation of task-specific information. In contrast, the baseline exhibits a sharper decay, suggesting that it fails to preserve meaningful structure during optimization.

The spectral analysis also provides insight into the variance and alignment improvements observed earlier. A smoother singular value distribution implies that perturbations are better aligned with the intrinsic geometry of the parameter space. This reduces the likelihood of exploring irrelevant directions, thereby lowering variance. At the same time, it increases the probability that updates align with true descent directions, leading to improved gradient alignment. Thus, the spectral behavior serves as a unifying explanation for multiple empirical observations.

Overall, the mechanism analysis confirms that the effectiveness of RoZO is rooted in its geometric formulation. By operating within the tangent space of a low-rank manifold, the method reshapes the optimization landscape in a way that favors stable and meaningful updates. This structural advantage distinguishes RoZO from existing zeroth-order methods, which rely on random perturbations and therefore fail to capture the underlying geometry of low-rank adaptation.