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

Zichen Song
Lanzhou University
Sungkyunkwan University
songzch21@lzu.edu.cn, sls530@skku.edu

Weijia Li
Lanzhou University
forgetfuljre@gmail.com

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 and 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, 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

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1773–1783

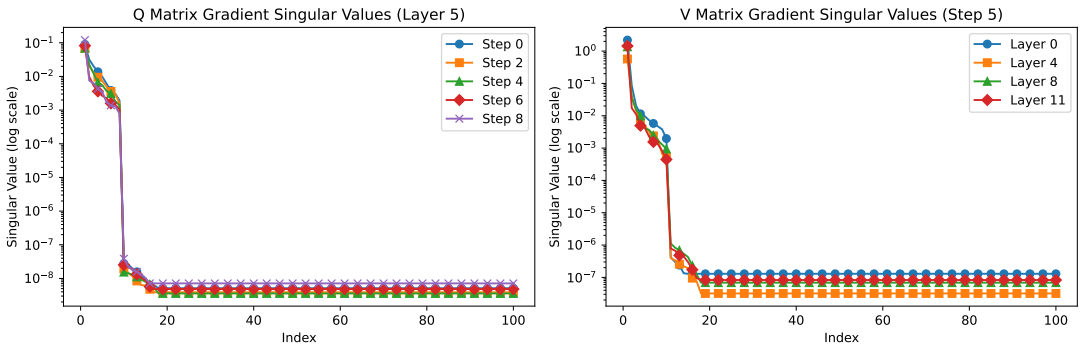

Figure 1: **Singular value distributions of attention gradients under RoZO. Left**: Singular value distribution of the gradient of the attention Q matrix in layer 5 across different fine-tuning steps. **Right**: Singular value distribution of the gradient of the attention V matrix across different layers 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.

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 and 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 and 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 $\boldsymbol{X} = \{X_\ell\}_{\ell=1}$, where $X_\ell \in \mathbb{R}^{m_\ell \times n_\ell}$ represents the weight matrix of the ℓ-th layer and 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 and Spokoiny, 2017):

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

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

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).

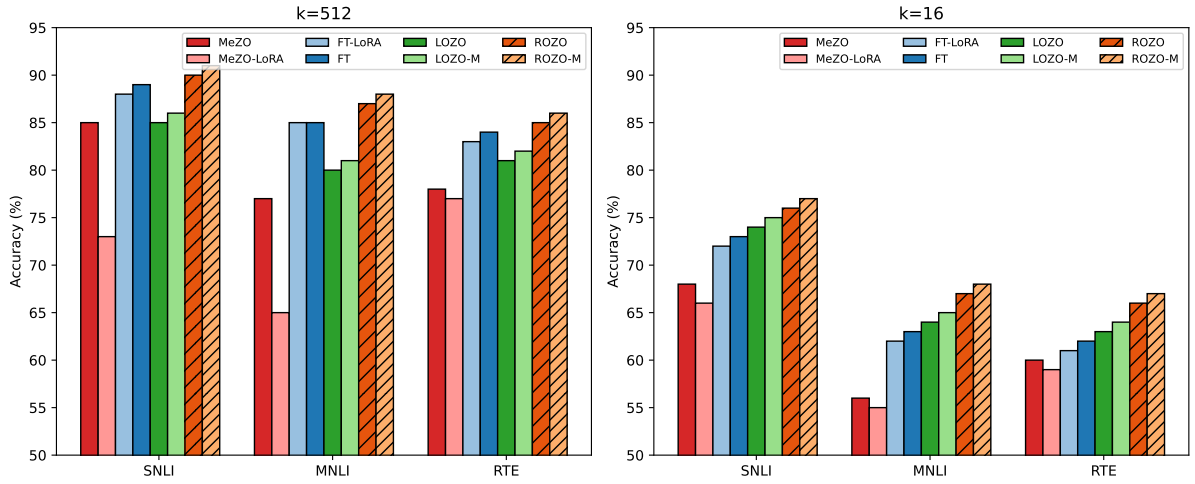

Figure 2: **Comparison of different fine-tuning methods on RoBERTa-large across three NLI tasks (SNLI, MNLI, and RTE).** The left panel corresponds to $k = 512$ training samples, and the right panel corresponds to $k = 16$ samples. While MeZO and MeZO-LoRA show limited accuracy and FT/FT-LoRA serve as 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.

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.

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

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

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,

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.

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.

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

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.

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.

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 $\sim$2.8 GB memory usage—significantly lower than MeZO-M (5.89

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*.

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 overparameterization.

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.

7 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.

8 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.

Limitations

While RoZO demonstrates strong performance across diverse datasets and model sizes, several limitations remain. First, the computational cost of zeroth-order optimization still grows with the number of queries q, and although our geometry-aware estimator reduces variance, fine-tuning very large-scale models (e.g., beyond 100B parameters) may remain challenging without additional acceleration techniques. Second, RoZO has primarily been evaluated on classification and QA tasks; its effectiveness on generative tasks, such as long-form text generation or code synthesis, remains to be fully explored. Third, while our experiments show promising scalability, hyperparameter tuning remains sensitive, particularly for perturbation radius and manifold rank, which may limit ease of adoption in practice. Addressing these limitations through automated tuning and further theoretical analysis is an important direction for future work.

Ethical Considerations

Our work focuses on improving the efficiency of fine-tuning large language models under black-box or memory-constrained settings. While RoZO itself does not introduce new data or model biases, any fine-tuned model inherits the ethical concerns of the underlying LLMs, including potential biases, misinformation propagation, or misuse in harmful applications. By reducing resource barriers to fine-tuning, RoZO may inadvertently enable wider deployment of powerful LLMs without sufficient oversight. We encourage responsible use of this method in alignment with community standards, including transparency about training data, careful evaluation of societal impacts, and the incorporation of safeguards to mitigate potential harms.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

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

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

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

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

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

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

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

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

Tanmay Gautam, Youngsuk Park, Hao Zhou, Parameswaran Raman, and Wooseok Ha. 2024. Variance-reduced zeroth-order methods for fine-tuning language models. *arXiv preprint arXiv:2404.08080*.

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

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

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

Yongchang Hao, Yanshuai Cao, and Lili Mou. 2024. Flora: Low-rank adapters are secretly gradient compressors. *arXiv preprint arXiv:2402.03293*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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

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

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

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

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

Zeman Li, Xinwei Zhang, and Meisam Razaviyayn. 2024. 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*.

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

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

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

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

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

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

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

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. 2017. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 164–169, New York, NY, USA. ACM.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, and 1 others. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and 1 others. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

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

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

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

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D Lee, Wotao Yin, Mingyi Hong, and 1 others. 2024. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark. *arXiv preprint arXiv:2402.11592*.

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

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

Pu Zhao, Sijia Liu, Pin-Yu Chen, Nghia Hoang, Kaidi Xu, Bhavya Kailkhura, and Xue Lin. 2019. 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*, pages 121–130.

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