ZEROTH-ORDER RIEMANNIAN OPTIMIZATION ON FIXED-RANK MANIFOLDS FOR LLM FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Zeroth-order (ZO) optimization estimates descent directions from function values alone, which removes the activation storage required by backpropagation and makes it attractive for memory-efficient fine-tuning of large language models (LLMs). Existing low-rank ZO methods either parameterize the additive update through Euclidean factors with a non-unique representation, or restrict each finite-difference direction to be low-rank. In this work, we propose to constrain each additive update block to a fixed-rank matrix manifold while keeping the pretrained weights frozen, so the learned update remains rank-constrained throughout fine-tuning. On this geometry, we build a Riemannian ZO gradient estimator that projects a random matrix onto the tangent space, normalizes it to correct the scale mismatch induced by projection, and further propose Zeroth-order Riemannian Gradient Descent (ZO-RGD) and its smoothed variant, ZO-RGD-S. We show that the ambient finite difference matches its retracted counterpart, the normalized estimator recovers the Riemannian gradient in expectation, and ZO-RGD converges under trajectory-regularity conditions. Experiments on LLM fine-tuning tasks show that the proposed algorithms are competitive with strong forward-only baselines across model families and tasks, with gains in several settings.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable success across a wide range of domains (Brown et al., 2020; Achiam et al., 2023). Fine-tuning these models on downstream tasks is crucial for their practical deployment. Parameter-efficient fine-tuning (PEFT) techniques such as low-rank adaptation (LoRA) (Hu et al., 2022) have emerged as a practical solution, training low-rank updates while keeping pretrained weights frozen (Hu et al., 2022; Liu et al., 2024a; Zhang et al., 2023; Hayou et al., 2024; Li et al., 2025; Yang et al., 2024a). However, first-order optimization still incurs substantial memory overhead from storing activations for backpropagation, especially in long-context settings where activations dominate memory. To enhance memory efficiency, zeroth-order (ZO) fine-tuning methods (Malladi et al., 2023; Chen et al., 2024; Zhang et al., 2024) estimate descent directions from function values without backpropagated gradients, avoiding the activation storage required by backward passes (Spall, 1992).

Recent low-rank ZO fine-tuning methods combine these two directions by restricting the finite-difference search through low-rank perturbations, adapter factors, gradient estimators, or random subspaces (Chen et al., 2025; Yu et al., 2025; Yang et al., 2024b; Song & Li, 2026). Factor-based approaches constrain the learned update through adapter coordinates, whereas low-rank perturbation methods constrain individual ZO steps. These two branches of methods differ in the way of constructing low-rank additive update matrices when optimizing a fixed-rank update matrix. Factor-based adaptation uses a factorized update $\Delta \boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$ and optimizes over $(\boldsymbol{B}, \boldsymbol{A})$ (Hu et al., 2022). This representation is nonunique, since $(\boldsymbol{B}\boldsymbol{Q})(\boldsymbol{Q}^{-1}\boldsymbol{A})$ gives the same update for any invertible $\boldsymbol{Q}$. Low-rank perturbation and random-subspace methods instead use restricted finite-difference directions or estimators to construct each ZO step (Chen et al., 2025; Yu et al., 2025; Yang et al., 2024b). For example, low-rank perturbation methods form each step from a sampled rank-r direction $\boldsymbol{U}\boldsymbol{V}^\top$, while random-subspace variants restrict the step direction to a prescribed low-dimensional subspace (Chen et al., 2025; Yu et al., 2025). As a result, factor-based methods optimize a non-unique parameterization, while perturbation and subspace methods restrict each ZO search direction.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To tackle these issues, we propose to optimize $\Delta \boldsymbol{W}$ as a fixed-rank matrix variable formulated on a Riemannian manifold. Consider one selected weight matrix with frozen pretrained value $\boldsymbol{W}_0$. At iteration t, the adapted weight is written as $\boldsymbol{W}_t = \boldsymbol{W}_0 + \Delta \boldsymbol{W}_t$, with

$$\Delta \boldsymbol{W}_t \in \mathcal{M}_r^{m \times n}, \qquad t = 0, \ldots, T, \tag{1}$$

where $\mathcal{M}_r^{m \times n}$ denotes the manifold of rank-r matrices in $\mathbb{R}^{m \times n}$, with $r \ll \min\{m, n\}$. Thus, for this block, the update trajectory stays on the same fixed-rank manifold. This turns fixed-rank structure from a property of an individual ZO step into a constraint on the persistent update trajectory. Building on this fixed-rank update trajectory, we propose a forward-only Riemannian ZO fine-tuning framework. At each step, we project a random matrix onto the current tangent space, normalize it, and retract the update back to the rank-r manifold to build the ZO Riemannian gradient. We use this gradient to design Zeroth-order Riemannian Gradient Descent (ZO-RGD) and a smoothed variant (ZO-RGD-S). Figure 1 illustrates the tangent random matrix and the retracted update for one representative matrix block. We summarize the contributions as follows:

1. We formulate forward-only LLM fine-tuning as optimization of persistent additive updates on fixed-rank matrix manifolds. For each selected matrix block, the update trajectory is maintained on a prescribed rank-r manifold, so the rank constraint is imposed on the optimization state rather than only on an individual ZO step.
2. We show that tangent projection makes the finite-difference perturbation scale depend on the rank and layer shape, and correct this mismatch with a normalized tangent estimator that needs only two forward passes and no additional retractions per step. Building on this estimator, we develop ZO-RGD and its smoothed variant ZO-RGD-S, which recursively smooths the finite-difference scalar at the same forward cost.
3. We prove that the implemented ambient finite difference agrees with its retracted counterpart to leading order, and that the normalized estimator recovers the Riemannian gradient in expectation up to an $\mathcal{O}(\epsilon^2)$ bias. Building on these properties, we establish a nonasymptotic convergence guarantee for ZO-RGD, conditional on regularity of the realized trajectory.
4. We evaluate the proposed methods across model families and downstream tasks against existing forward-only baselines. The results show competitive performance, with gains in several settings.

2 Related Work

Low-rank adaptation. LoRA adapts a frozen pretrained model by adding a low-rank update to selected weight matrices (Hu et al., 2022). This idea has led to a large family of PEFT methods, including weight-decomposed adaptation (Liu et al., 2024a), adaptive rank allocation and learning-rate asymmetry (Zhang et al., 2023; Hayou et al., 2024), nonzero initialization and analyses of low-rank training dynamics (Li et al., 2025; Yaras et al., 2024), tensorized update parameterizations (Yang et al., 2024c), rank-one or high-rank structured variants (Sheng et al., 2025; Huang et al., 2025), low-rank gradient projection (Zhao et al., 2024a), and broader PEFT surveys (Yang et al., 2024a). These methods establish low-rank updates as an effective adaptation object. In a forward-only ZO setting, we ask a different question: whether the update matrix itself can be the constrained optimization state, rather than representing it through nonunique factor coordinates.

Zeroth-order fine-tuning. Classical ZO theory studies simultaneous perturbation, Gaussian or spherical smoothing, and two-point randomized estimators (Spall, 1992; Ghadimi & Lan, 2013; Duchi et al., 2015; Nesterov & Spokoiny, 2017; Shamir, 2017; Berahas et al., 2022; Balasubramanian & Ghadimi, 2022). MeZO showed that such forward-only estimates can fine-tune LLMs while avoiding activation-gradient storage (Malladi et al., 2023). Subsequent work has expanded the optimizer and benchmark landscape through large-scale studies (Chen et al., 2024), benchmark-oriented analyses (Zhang et al., 2024), random-subspace or low-rank perturbation methods (Chen et al., 2025; Yu et al., 2025; Yang et al., 2024b), and variants that use curvature or projected-alignment information (Zhao et al., 2024b; Wang et al., 2026; Mi et al., 2025). Other approaches explore sparse perturbations (Liu et al., 2024b) and step-size adaptation based on batch-loss variability (Dang et al., 2025). LOZO (Chen et al., 2025) is the closest predecessor because it introduces low-rank structure into ZO estimation for LLMs. The distinction is where the structure lives: LOZO forms a low-rank estimator or subspace update, whereas the proposed algorithms construct their random matrices in the tangent space of the current fixed-rank updates.

Riemannian optimization on fixed-rank matrix manifolds. Fixed-rank matrix manifolds, tangent projections, and retractions are standard tools for optimizing low-rank matrices without exposing redundant factor gauges (Absil et al., 2008; Vandereycken, 2013; Wei et al., 2016). Riemannian ZO optimization studies stochastic optimization over Riemannian submanifolds embedded in Euclidean space when only noisy function evaluations are available (Li et al., 2022). Recent deep-learning work uses this geometry to optimize low-rank weights directly or to design Riemannian low-rank optimizers and preconditioners (Bian et al., 2025; Zhang & Pilanci, 2024; Bogachev et al., 2025; Mo et al., 2025). RoZO (Song & Li, 2026) is a closely related concurrent work that combines ZO optimization with geometry induced by low-rank adapter factors, while our method takes the matrix update itself as the optimization variable on the fixed-rank manifold. Riemannian ZO methods establish convergence for exponential-map or retraction-based finite differences under geometric and smoothness assumptions (Li et al., 2023; He et al., 2024). Structure-preserving metrics have also been proposed to address geodesic incompleteness (Ma & Huang, 2026).

3 Preliminaries

3.1 Zeroth-order optimization

For a stochastic objective $f(\boldsymbol{X}) = \mathbb{E}_\xi[F(\boldsymbol{X};\xi)]$, ZO methods estimate descent directions from function values. Given a data realization ξ and a random matrix $\boldsymbol{Z}$ with i.i.d. standard Gaussian entries, the two-point estimator is

$$\widehat{\nabla} F(\boldsymbol{X};\xi) = \frac{F(\boldsymbol{X}+\epsilon\boldsymbol{Z};\xi) - F(\boldsymbol{X}-\epsilon\boldsymbol{Z};\xi)}{2\epsilon}\,\boldsymbol{Z}, \tag{2}$$

where $\epsilon > 0$ controls the perturbation radius (Duchi et al., 2015; Nesterov & Spokoiny, 2017; Shamir, 2017; Berahas et al., 2022; Balasubramanian & Ghadimi, 2022). The scalar difference estimates a directional derivative, together with $\boldsymbol{Z}$, gives a randomized gradient estimate. MeZO uses two forward passes and regenerates $\boldsymbol{Z}$ from seeds to avoid backpropagation and storing perturbations (Malladi et al., 2023).

3.2 Fixed-rank manifold geometry

For a matrix block, we can define the fixed-rank manifold as

$$\mathcal{M}_r^{m\times n} = \{\boldsymbol{X} \in \mathbb{R}^{m\times n} : \operatorname{rank}(\boldsymbol{X}) = r\}, \qquad d_r = r(m+n-r), \tag{3}$$

with the Frobenius metric $\langle \boldsymbol{G}_1, \boldsymbol{G}_2\rangle_{\mathrm{F}} = \operatorname{tr}(\boldsymbol{G}_1^\top \boldsymbol{G}_2)$. We write $\mathcal{M}_r$ when the block dimensions are clear. At $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, the thin-SVD factors $\boldsymbol{U}, \boldsymbol{V}$ have orthonormal columns and $\boldsymbol{\Sigma}$ contains positive singular values. The Frobenius-orthogonal projection of a matrix $\boldsymbol{Z}$ onto the tangent space $T_{\boldsymbol{X}}\mathcal{M}_r$ is (Absil et al., 2008; Vandereycken, 2013; Wei et al., 2016) denoted as

$$P_{T_{\boldsymbol{X}}\mathcal{M}_r}(\boldsymbol{Z}) = \boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^\top - \boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^\top. \tag{4}$$

The remaining matrix is an admissible first-order direction at the current update, and the tangent space changes as that update moves. For a step $\boldsymbol{Y} \in T_{\boldsymbol{X}}\mathcal{M}_r$, we can return it to $\mathcal{M}_r$ using the rank-r truncated-SVD map $\mathcal{H}_r$:

$$\mathcal{R}_{\boldsymbol{X}}(\boldsymbol{Y}) = \mathcal{H}_r(\boldsymbol{X}+\boldsymbol{Y}). \tag{5}$$

The map retains the leading r singular triplets and gives the closest rank-r matrix in Frobenius distance locally. It is a smooth local retraction on suitable neighborhoods away from the rank-deficient boundary (Absil et al., 2008; Vandereycken, 2013; Bian et al., 2025).

4 Proposed method

4.1 Problem formulation

Let $\mathcal{S}$ index the selected matrix blocks, with frozen pretrained values $\boldsymbol{W}_{0,\ell}$ and learned additive updates $\Delta\boldsymbol{W}_\ell \in \mathcal{M}_{r_\ell}^{m_\ell\times n_\ell}$. The effective weights are $\boldsymbol{W}_{0,\ell} + \Delta\boldsymbol{W}_\ell$. Writing $\mathcal{E} = \prod_{\ell\in\mathcal{S}} \mathbb{R}^{m_\ell\times n_\ell}$,

the matrix updates lie on the product manifold:

$$\mathcal{M} = \prod_{\ell \in \mathcal{S}} \mathcal{M}_{r_\ell}^{m_\ell \times n_\ell} \subset \mathcal{E}. \tag{6}$$

Trainable one-dimensional parameters, including biases and LayerNorm parameters, form a Euclidean factor $\boldsymbol{b} \in \mathbb{R}^{d_0}$, while all unselected parameters remain frozen. Thus the full trainable state belongs to the product manifold $\mathcal{X} = \mathcal{M} \times \mathbb{R}^{d_0}$. Unlike Euclidean factor coordinates, this representation is used only to compute tangent projections and retractions of the matrix variable, and the method is invariant to the choice of representative (see Appendix A.1). The loss function is defined on the surrounding Euclidean space:

$$F(\Delta \boldsymbol{W}, \boldsymbol{b}; \xi) := \mathcal{L}(\boldsymbol{W}_0 + \Delta \boldsymbol{W}, \boldsymbol{b}; \xi), \quad (\Delta \boldsymbol{W}, \boldsymbol{b}) \in \mathcal{E} \times \mathbb{R}^{d_0}. \tag{7}$$

It couples all blocks and need not decompose over them. We propose to minimize its expectation formulated on the product manifold $\mathcal{X}$ subject to the mixed constraint:

$$\min_{(\Delta \boldsymbol{W}, \boldsymbol{b}) \in \mathcal{X}} f(\Delta \boldsymbol{W}, \boldsymbol{b}) := \mathbb{E}_\xi[F(\Delta \boldsymbol{W}, \boldsymbol{b}; \xi)]. \tag{8}$$

4.2 ZO Riemannian Gradient Estimation

We first describe one representative matrix block, suppressing its index. In the full model, independent blockwise random matrices are applied jointly to the coupled loss, and two forward evaluations produce one scalar coefficient shared by all updates.

Finite-difference scale mismatch: At $\Delta \boldsymbol{W}_t \in \mathcal{M}_r$, we draw a Gaussian matrix $\boldsymbol{Z}_t \in \mathbb{R}^{m \times n}$ with i.i.d. standard normal entries and project it using the orthogonal projection in (4):

$$\widehat{\boldsymbol{Z}}_t = P_{T_{\Delta \boldsymbol{W}_t} \mathcal{M}_r}(\boldsymbol{Z}_t). \tag{9}$$

This gives a valid tangent direction, but changes the scale of the finite differences. The ambient Gaussian satisfies $\|\boldsymbol{Z}_t\|_{\mathrm{F}}^2 \sim \chi^2_{mn}$, whereas its tangent projection has $\|\widehat{\boldsymbol{Z}}_t\|_{\mathrm{F}}^2 \sim \chi^2_{d_r}$, where $d_r = r(m+n-r)$. In the low-rank regime, $d_r \ll mn$, so a fixed ϵ yields a typical perturbation radius $\epsilon\sqrt{d_r}$ instead of $\epsilon\sqrt{mn}$. Consequently, changing the rank or layer shape also changes the perturbation scale, even when the nominal finite-difference parameter remains fixed.

Normalized gradient estimator: To remove this dependence of the perturbation scale on the rank and layer shape, we decouple the scale from the tangent projection by normalizing $\widehat{\boldsymbol{Z}}_t$:

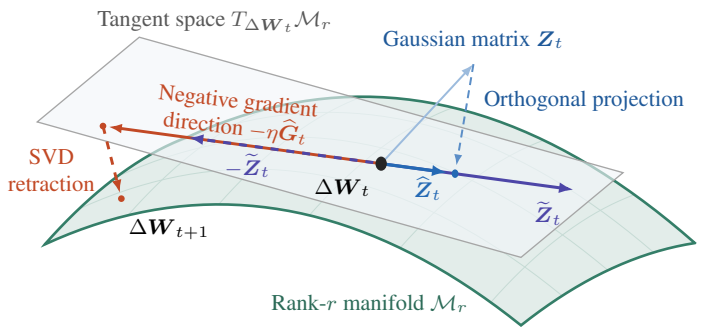
$$\widetilde{\boldsymbol{Z}}_t = \sqrt{mn}\, \frac{\widehat{\boldsymbol{Z}}_t}{\|\widehat{\boldsymbol{Z}}_t\|_{\mathrm{F}}}. \tag{10}$$

The normalized random matrix remains in $T_{\Delta \boldsymbol{W}_t} \mathcal{M}_r$ and has fixed norm $\sqrt{mn}$, matching the standard ambient Gaussian scale while retaining the tangent-space geometry. Given the same sampled realization ξ_t for both evaluations, we form

$$\hat{s}_t = \frac{F(\Delta \boldsymbol{W}_t + \epsilon \widetilde{\boldsymbol{Z}}_t; \xi_t) - F(\Delta \boldsymbol{W}_t - \epsilon \widetilde{\boldsymbol{Z}}_t; \xi_t)}{2\epsilon}, \tag{11}$$

$$\widehat{\boldsymbol{G}}_t = \hat{s}_t \widetilde{\boldsymbol{Z}}_t \in T_{\Delta \boldsymbol{W}_t} \mathcal{M}_r. \tag{12}$$

The scalar $\hat{s}_t$ estimates the directional derivative of the sampled loss along $\widetilde{\boldsymbol{Z}}_t$, and multiplying it by $\widetilde{\boldsymbol{Z}}_t$ yields a Riemannian gradient estimator on the tangent space. In the product setting, the two evaluations of F perturb all selected blocks jointly rather than one block at a time. Although $\widetilde{\boldsymbol{Z}}_t$ lies in the current tangent space, the evaluation points $\Delta \boldsymbol{W}_t \pm \epsilon \widetilde{\boldsymbol{Z}}_t$ in (11) generally do not lie on $\mathcal{M}_r$. These temporary points are nevertheless admissible, because the loss in (7) is defined on the ambient space and only the persistent update is returned to the manifold by retraction. This choice saves two retractions per step. In Section 5, we will show that it agrees with retracted finite differences to leading order (Proposition 1), and the normalized estimator (12) remains aligned in expectation with the Riemannian gradient (Proposition 2).



Normalized random matrix

$\widetilde{\boldsymbol{Z}}_t = \sqrt{mn}\,\widehat{\boldsymbol{Z}}_t / \|\widehat{\boldsymbol{Z}}_t\|_{\mathrm{F}}$

Two forward evaluations

$F_{\pm} = F(\Delta \boldsymbol{W}_t \pm \epsilon \widetilde{\boldsymbol{Z}}_t; \xi_t)$

$\hat{s}_t = \dfrac{F_+ - F_-}{2\epsilon}$

Update and retract

$\widehat{\boldsymbol{G}}_t = \hat{s}_t \widetilde{\boldsymbol{Z}}_t$

$\Delta \boldsymbol{W}_{t+1} = \mathcal{R}_{\Delta \boldsymbol{W}_t}(-\eta \widehat{\boldsymbol{G}}_t)$

Figure 1: **Two loss evaluations determine a fixed-rank ZO-RGD update.** The orthogonal projection onto $T_{\Delta \boldsymbol{W}_t}\mathcal{M}_r$ gives $\widehat{\boldsymbol{Z}}_t = P_{T_{\Delta \boldsymbol{W}_t}\mathcal{M}_r}(\boldsymbol{Z}_t)$, and normalization gives $\widetilde{\boldsymbol{Z}}_t = \sqrt{mn}\,\widehat{\boldsymbol{Z}}_t / \|\widehat{\boldsymbol{Z}}_t\|_{\mathrm{F}}$. The two points $\Delta \boldsymbol{W}_t \pm \epsilon \widetilde{\boldsymbol{Z}}_t$ along these directions jointly perturb all selected blocks to obtain $F_{\pm}$. Their difference gives the directional derivative $\hat{s}_t$ and thus the zeroth-order Riemannian gradient $\widehat{\boldsymbol{G}}_t$. Truncated SVD retracts $\Delta \boldsymbol{W}_t - \eta \widehat{\boldsymbol{G}}_t$ back to the rank-r manifold.

Algorithm 1 ZO-RGD

Require: Frozen weights $\boldsymbol{W}_0$; initial update $\Delta \boldsymbol{W}_0 \in \mathcal{M}$; learning rate η; perturbation radius ϵ.

1: **for** $t = 0, 1, 2, \ldots$ **do**
2: Sample a realization ξ_t, Gaussian random matrices for all selected blocks, and $\boldsymbol{z}_{0,t}$.
3: Project and normalize each random matrix using (9)–(10).
4: Compute $\hat{s}_t$ with two joint forward evaluations as in (11).
5: Form the ZO Riemannian gradient $\widehat{\boldsymbol{G}}_{t,\ell} = \hat{s}_t \widetilde{\boldsymbol{Z}}_{t,\ell}$ for $\ell \in \mathcal{S}$ as in (12).
6: Retract each block as in (13) and set $\boldsymbol{b}_{t+1} = \boldsymbol{b}_t - \eta \hat{s}_t \boldsymbol{z}_{0,t}$.
7: **end for**

4.3 ZO Riemannian Optimization Algorithms

As illustrated in Figure 1, the estimator in (12) provides a search direction in the current tangent space, but a step along it is located on the tangent space. We now combine it with the retraction (5) to obtain two algorithms on the product manifold in (6): ZO-RGD, the basic retracted gradient step, and ZO-RGD-S, which adds scalar smoothing at the same cost of two forward passes per iteration.

ZO-RGD: ZO-RGD turns the normalized tangent estimator into a forward-only update. After the two joint evaluations determine $\hat{s}_t$, each matrix block takes a step along its own random matrix. The fixed-rank retraction (5) then maps the trial point back to the constraint set:

$$\Delta \boldsymbol{W}_{t+1,\ell} = \mathcal{R}_{\Delta \boldsymbol{W}_{t,\ell}}\big(-\eta \widehat{\boldsymbol{G}}_{t,\ell}\big), \qquad \ell \in \mathcal{S}. \tag{13}$$

The same scalar $\hat{s}_t$ drives every matrix block, while the tangent spaces and retractions remain block-specific. The Euclidean factor takes the plain step $\boldsymbol{b}_{t+1} = \boldsymbol{b}_t - \eta \hat{s}_t \boldsymbol{z}_{0,t}$ along its raw Gaussian vector $\boldsymbol{z}_{0,t} \in \mathbb{R}^{d_0}$. We initialize the update (13) at a near-zero fixed-rank point (see Appendix A.2), which yields the complete sequence summarized in Algorithm 1.

ZO-RGD with smoothing: ZO-RGD-S uses the same random matrices and two forward evaluations, but recursively smooths the scalar coefficient in a fixed matrix-block order $\ell_1, \ldots, \ell_K$:

$$m_{t,k} = \beta m_{t,k-1} + (1-\beta)\hat{s}_t, \qquad k = 1, \ldots, K, \tag{14}$$

$$\widehat{\boldsymbol{G}}^{\mathrm{S}}_{t,\ell_k} = m_{t,k} \widetilde{\boldsymbol{Z}}_{t,\ell_k}, \tag{15}$$

$$\Delta \boldsymbol{W}_{t+1,\ell_k} = \mathcal{R}_{\Delta \boldsymbol{W}_{t,\ell_k}}\big(-\eta \widehat{\boldsymbol{G}}^{\mathrm{S}}_{t,\ell_k}\big). \tag{16}$$

The scalar state passes between steps as $m_{t,0} = m_{t-1,K}$ for $t \geq 1$. On first use, we set $m_{0,0} = \hat{s}_0$, so the first-step coefficients equal those of ZO-RGD. For subsequent steps, the recursion progressively blends the previous state toward $\hat{s}_t$ along the fixed block order, so different blocks generally use different smoothed coefficients within the same iteration. Algorithm 2 in Appendix A.3 gives the full pseudocode.

Remark 1 (scalar smoothing). *ZO-RGD-S carries only the scalar $m_{t,k}$ between iterations and no matrix-valued state across tangent spaces. It is therefore scalar smoothing rather than transported Riemannian momentum, and it requires no vector transport between $T_{\Delta \boldsymbol{W}_{t,\ell}}\mathcal{M}_{r_\ell}$ and $T_{\Delta \boldsymbol{W}_{t+1,\ell}}\mathcal{M}_{r_\ell}$. Setting $\beta = 0$ gives $m_{t,k} = \hat{s}_t$ for every k and recovers ZO-RGD.*

Blockwise implementation and memory: Each matrix block stores a thin-SVD update with $\mathcal{O}((m_\ell + n_\ell)r_\ell)$ parameters, and ZO-RGD-S adds only the scalar smoothing state. Random matrices are regenerated from the same seed for both evaluations and the update, so dense perturbations need not be retained (Malladi et al., 2023; Chen et al., 2025). The projection and retraction (QR factorizations and an SVD of a $2r_\ell \times 2r_\ell$ core) act on dense tangent matrices and cost $\mathcal{O}(m_\ell n_\ell r_\ell)$ per block (see Appendix A.4).

5 Theoretical Results

This section supports the estimator and algorithm of Section 4 with three results: the ambient finite difference matches its retracted counterpart to leading order (Proposition 1), the normalized estimator recovers the Riemannian gradient in expectation up to a known scaling (Proposition 2), and ZO-RGD converges under trajectory-regularity conditions (Theorem 1). Proofs of these two propositions and one theorem are deferred to Appendices B and C, respectively.

Proposition 1 (Ambient and retracted finite differences). *Let $F(\cdot;\xi_t)$ be differentiable near $\Delta \boldsymbol{W}_t \in \mathcal{M}_r$, and let $\mathcal{R}_{\Delta \boldsymbol{W}_t}(\tau \boldsymbol{Y}) = \Delta \boldsymbol{W}_t + \tau \boldsymbol{Y} + \mathcal{O}(\tau^2)$ as $\tau \to 0$ for every tangent $\boldsymbol{Y}$. If $\hat{s}_t^{\mathrm{retr}}$ denotes (11) evaluated at $\mathcal{R}_{\Delta \boldsymbol{W}_t}(\pm\epsilon \widetilde{\boldsymbol{Z}}_t)$, then $\hat{s}_t = \hat{s}_t^{\mathrm{retr}} + o(1)$ as $\epsilon \to 0$.*

A retraction follows the tangent line to first order, so both finite differences share the same limiting directional derivative. At finite ϵ, they evaluate different points and can give different coefficients, therefore, the analysis below concerns the implemented ambient estimator.

Proposition 2 (Expected gradient alignment). *Fix $\Delta \boldsymbol{W}_t \in \mathcal{M}_r$ and ξ_t, and assume $F(\cdot;\xi_t)$ has a locally Lipschitz Hessian near $\Delta \boldsymbol{W}_t$. For the estimator in (12) and the Frobenius metric, as $\epsilon \to 0$,*

$$\mathbb{E}_{\boldsymbol{Z}_t}[\widehat{\boldsymbol{G}}_t \mid \Delta \boldsymbol{W}_t, \xi_t] = \frac{mn}{d_r} \operatorname{grad}_{\mathcal{M}_r} F(\Delta \boldsymbol{W}_t; \xi_t) + \mathcal{O}(\epsilon^2). \tag{17}$$

If $\mathbb{E}_{\xi_t}[F] = f$ and differentiation commutes with expectation, the same holds with f in place of F.

Proposition 2 shows that, in expectation, the estimator of each fixed-rank block is its Riemannian gradient scaled by $c_\ell = m_\ell n_\ell / d_{r_\ell} \geq 1$. Measuring stationarity in the correspondingly weighted norm $\|\cdot\|_{\boldsymbol{D}}$ (Appendix C.1), which dominates the plain gradient norm, we obtain the following convergence result, conditional on the trajectory regularity in Assumptions 1–3 (Appendix C.2). The formal statement with explicit stepsize and constants is in Appendix C.3.

Theorem 1 (Conditional convergence of ZO-RGD, informal). *Suppose Assumptions 1–3 hold along the realized trajectory. Then, with a constant stepsize $\eta = \mathcal{O}(T^{-1/2})$, ZO-RGD satisfies*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\big[\|\operatorname{grad} f(\boldsymbol{x}_t)\|_{\boldsymbol{D}}^2\big] = \mathcal{O}(T^{-1/2}) + \mathcal{O}(T^{-1}) + \mathcal{O}(\epsilon^4), \tag{18}$$

where the hidden constants depend on the initial gap, the smoothness and variance constants, and the trainable dimension.

6 Experiments

This section evaluates ZO-RGD and ZO-RGD-S as forward-only fine-tuning methods. Section 6.2 compares them against cited MeZO and LOZO results on the eleven-task OPT suite at 13B and 30B, the scale at which those baselines report their main results. Section 6.3 removes the reporting asymmetries of cited numbers with a self-run comparison in which every method shares the same protocol, checkpoint rule, and seeds. Section 6.4 tests whether the gains carry over to Llama-3.1-8B and Qwen2.5-7B. Section 6.5 isolates the geometric ingredients of the method through controlled ablations and reports the computational price of projection and retraction. Full configurations, per-seed values, the complete OPT-1.3B comparison, and further diagnostics are in Appendix D.

Table 1: OPT-13B main comparison on the LOZO eleven-task suite. The results of Zero-shot, ICL, MeZO, MeZO-LoRA, and LOZO are from Chen et al. (2025); ZO-RGD and ZO-RGD-S are evaluated under the setup described above, with each cell selected by the development metric within that variant's 20,000-step sweep cells. Best and second-best results in each column are shown in **bold** and underlined, respectively. Fail denotes that the ZO-RGD's ReCoRD run diverged and produced no valid evaluation (see Appendix D.18 for details).

Method	SST-2	RTE	CB	BoolQ	WSC	WiC	MultiRC	COPA	ReCoRD	SQuAD	DROP
Zero-shot	58.8	59.6	46.4	59.0	38.5	55.0	46.9	80.0	81.0	46.2	14.6
ICL	87.0	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	88.0	81.3	81.8	**31.3**
MeZO-LoRA	89.6	67.9	67.8	73.5	**63.5**	60.2	61.3	84.0	81.5	82.1	**31.3**
LOZO	91.7	70.4	69.6	71.9	**63.5**	60.8	63.0	**89.0**	81.3	84.9	30.7
ZO-RGD	92.43	**72.20**	66.07	71.40	59.62	**62.54**	57.80	87.00	Fail	**87.19**	30.02
ZO-RGD-S	**92.55**	70.04	**71.43**	**74.20**	61.54	61.13	**63.70**	85.00	82.10	84.06	28.03

Table 2: OPT-30B scaling comparison on the LOZO task subset. Zero-shot, ICL, MeZO, and LOZO are cited from Chen et al. (2025); ZO-RGD and ZO-RGD-S are evaluated with the same training budget, with checkpoint-selection details reported in Appendix D.1. Best and second-best results in each column are shown in **bold** and underlined, respectively. SQuAD uses F1, and the other columns use the OPT task metrics.

Method	SST-2	RTE	BoolQ	WSC	WiC	SQuAD
Zero-shot	56.7	52.0	39.1	38.5	50.2	46.5
ICL	81.9	66.8	66.2	56.7	51.3	78.0
MeZO	90.7	64.3	68.2	63.5	56.3	**86.1**
LOZO	92.8	65.3	**72.3**	**64.4**	57.2	85.6
ZO-RGD	92.89	68.59	70.50	63.46	57.68	84.89
ZO-RGD-S	**94.04**	**71.12**	**72.30**	61.54	**58.15**	85.11

6.1 Setup

We evaluate the fixed-rank update formulation on OPT-1.3B, OPT-13B, and OPT-30B (Zhang et al., 2022), using the eleven-task MeZO/LOZO suite and its task metrics (Malladi et al., 2023; Chen et al., 2025; Wang et al., 2019; Rajpurkar et al., 2016; Dua et al., 2019), and test portability on Llama-3.1-8B and Qwen2.5-7B. Our runs use 1000 training examples and 20,000 updates, and each update uses two forward passes.

6.2 Broad-suite comparison with cited baselines

OPT-13B. Table 1 follows the OPT-13B table format of LOZO. The cited rows are kept in the same order, and ZO-RGD and ZO-RGD-S are added under the setup above. This is the main large-model comparison in the paper because it uses the full eleven-task OPT suite at the scale where MeZO and LOZO report their main OPT results.

Against the strongest cited ZO training row among MeZO, MeZO-LoRA, and LOZO, ZO-RGD-S improves on 6 of its 11 tasks and ZO-RGD on 4 of its 10 reported tasks. On 8 of the 11 tasks at least one of the two variants is above the strongest cited row: SST-2, RTE, CB, BoolQ, WiC, MultiRC, ReCoRD, and SQuAD, spanning sentiment classification, entailment, semantic reasoning, and extractive QA. WSC, COPA, and DROP remain stronger for the cited baselines, so the advantage is broad but not uniform across all tasks.

OPT-30B. Table 2 follows the OPT-30B part of the LOZO scaling table. We keep the zero-shot, ICL, MeZO, and LOZO rows from the cited table and add ZO-RGD and ZO-RGD-S. This table evaluates whether the same fixed-rank additive-update formulation remains effective at OPT-30B scale. Because checkpoint selection is not fully matched for OPT-30B, we use this table as scaling evidence, with run-level reporting details given in Appendix D.1.

At OPT-30B, the better of ZO-RGD and ZO-RGD-S improves over the cited LOZO row on SST-2, RTE, and WiC, ties LOZO on BoolQ, and remains competitive on WSC and SQuAD. Together with the OPT-13B results, this scaling study supports the central method claim: fixed-rank constraints

Table 3: Controlled OPT comparison with fixed report-seed hyperparameters and symmetric development-loss checkpoint selection. OPT-13B: mean ± sd over three seeds (RTE: accuracy; SQuAD: F1). OPT-30B BoolQ: single-seed accuracy at learning rate 10^{-7}. ZO-RGD uses $r = 4$, ZO-RGD-S uses $r = 2$. Passes/update counts forward (F) and backward (B) evaluations; bold marks the best forward-only result. NR denotes a matched result not reported in this comparison; OOM denotes an out-of-memory failure.

		OPT-13B		OPT-30B
Method	Passes/update	RTE	SQuAD	BoolQ
MeZO	2F	60.4 ± 1.5	82.8 ± 1.0	66.80
HiZOO[†]	4F	62.1 ± 2.6	NR	NR
LOZO	2F	62.1 ± 0.6	82.2 ± 0.7	69.40
ZO-RGD	2F	61.9 ± 2.9	83.3 ± 1.0	**72.70**
ZO-RGD-S	2F	$\mathbf{69.4 \pm 3.3}$	$\mathbf{84.9 \pm 1.3}$	72.50
LoRA+AdamW[‡]	1F+1B	83.4 ± 2.9	OOM	NR

[†] HiZOO uses the released implementation at learning rate 10^{-7} and completed RTE seeds $\{0, 1, 2\}$; the other 13B rows use seeds $\{1, 2, 3\}$. [‡] LoRA+AdamW is a first-order reference and is excluded from the forward-only ranking.

Table 4: Modern-model results (mean ± sd, three seeds, %; RTE/BoolQ: accuracy; SQuAD: F1; GSM8K: exact match on n=200).

	Llama-3.1-8B				Qwen2.5-7B			
Method	RTE	BoolQ	SQuAD	GSM8K	RTE	BoolQ	SQuAD	GSM8K
LOZO	56.9 ± 7.1	61.6 ± 2.7	86.8 ± 2.2	31.2 ± 2.4	86.6 ± 1.9	80.7 ± 0.6	87.0 ± 0.8	74.2 ± 2.8
ZO-RGD	62.5 ± 3.2	66.9 ± 1.5	87.1 ± 0.8	37.7 ± 5.8	88.3 ± 0.8	82.4 ± 1.2	87.9 ± 0.3	73.2 ± 2.8
ZO-RGD-S	62.9 ± 8.4	85.8 ± 1.1	85.5 ± 1.0	34.0 ± 1.0	88.3 ± 1.0	82.3 ± 0.5	87.6 ± 0.2	70.3 ± 2.8

can be imposed directly on learned additive updates while preserving the forward-only fine-tuning pattern. The complete local forward-only comparison at OPT-1.3B is in Appendix D.3, where the two Riemannian variants give the best forward-only result on 7 of 11 tasks.

6.3 Controlled comparison under a symmetric protocol

Table 3 complements the broad-suite results with self-run baselines, method-specific development-selected learning rates fixed across report seeds, and symmetric development-loss checkpoint selection. ZO-RGD-S reaches 69.4 ± 3.3 on RTE and 84.9 ± 1.3 on SQuAD, exceeding MeZO and LOZO on both OPT-13B tasks. The first-order LoRA+AdamW reference is stronger on RTE but runs out of A100-80G memory on SQuAD at batch size 16 without activation checkpointing. The OPT-30B BoolQ column is a single-seed spot-check under the same symmetric protocol (see Appendix D.6). Per-seed values and a shared-learning-rate regime, including loss trajectories against forward passes, are in Appendices D.7 and D.9.

Scalar smoothing at matched rank. At matched rank ($r = 2$) and learning rate on OPT-13B (Appendix D.8, Table 14), ZO-RGD-S has a higher mean than ZO-RGD on RTE (69.4 versus 64.5), while ZO-RGD has a slightly higher mean on SQuAD (85.4 versus 84.9). Neither paired difference is statistically significant (two-sided paired t-tests, $p = 0.28/0.47$); these three-seed comparisons do not establish an advantage for either variant.

6.4 Portability beyond OPT

Table 4 extends the recipe to Llama-3.1-8B and Qwen2.5-7B across four tasks. Results support portability across model families, with task-dependent performance. Learning rates come from method-specific development pilots. A seed-0 Llama BoolQ learning-rate diagnostic (Appendix D.10, Table 17) shows a strong learning-rate effect for all three methods: at 10^{-7}, LOZO reaches 80.9, and both Riemannian variants reach 85.5. This single-seed observation is separate from the three-seed panel and does not establish a method ranking.

Table 5: OPT-1.3B geometry controls (%; mean ± sd, three paired seeds). Left: finite-difference geometry on SST-2. Right: manifold versus BA-factor ZO with matched budgets (accuracy; SQuAD: F1). Learning rates are selected separately for each panel; compare within a panel. Right-panel rates use a shared 2,000-step development pilot: $10^{-6}/10^{-7}/10^{-6}$ for the manifold arm on SST-2/RTE/SQuAD and 10^{-4} for BA-factor.

Finite-difference rule	SST-2
Tangent-projected, normalized	92.4 ± 0.2
Retracted evaluation points	89.6 ± 0.9
Ambient rank-r outer product	54.3 ± 1.1
Outer product + LR compensation	64.7 ± 3.9

Task	Manifold	BA-factor
SST-2	87.7 ± 0.6	78.5 ± 10.0
RTE	56.0 ± 1.4	58.4 ± 4.0
SQuAD	71.3 ± 0.5	67.7 ± 1.3

Table 6: Measured cost at OPT-13B SQuAD, averaged over the two A100-80G hardware repeats in Table 23. LoRA+AdamW exhausts the A100 on SQuAD (batch 16, no activation checkpointing); its 126.2 GB entry is a separate three-run RTE mean on H200.

Method	Passes/update	Time/step (s)	Peak reserved (GB)
MeZO	2F	2.03	39.6
LOZO	2F	1.93	39.4
ZO-RGD	2F	3.51	39.4
ZO-RGD-S	2F	3.54	39.4
LoRA+AdamW	1F+1B	OOM	126.2 (H200, RTE)

6.5 Geometry and computational cost

Geometry and parameterization. The controls in Table 5 test the finite-difference rule and the optimization variable in separate blocks. Normalized tangent finite differences outperform the tested alternatives on SST-2, including learning-rate-compensated ambient random matrices. The separate parameterization comparison favors the manifold update on SST-2 and SQuAD, but not RTE at the pilot-selected rates. A separate three-seed SST-2 check at the manifold arm's 20,000-step grid rate 10^{-7} gives 92.7 ± 0.7, illustrating sensitivity to the tuning budget. Further controls, negative results, ablations, and stability diagnostics are in Appendices D.11–D.15.

Runtime and memory. Across two OPT-13B SQuAD profiles on different A100-80G cards (1,000 steps per method per card), ZO-RGD/ZO-RGD-S average $3.51/3.54$ seconds per update versus LOZO's 1.93, all at approximately 39.4 GB peak reserved memory as shown in Table 6. Thus comparable memory comes with higher per-step cost. Per-step times differ by less than 1% between the two hardware repeats, and peak reserved memory agrees at the logged precision (see Table 23, Appendix D.16). LoRA+AdamW runs out of memory at its first forward pass in both allocations under the same batch-size-16 configuration; the fixed-rank update state itself is negligible (see Appendix D.17).

7 Conclusion

We formulate forward-only LLM fine-tuning as optimization of persistent additive updates on fixed-rank matrix manifolds, so that the rank constraint is imposed on the optimization state rather than on individual ZO steps. ZO-RGD estimates the Riemannian gradient from two forward passes with normalized tangent finite differences and retracts each update onto the manifold. ZO-RGD-S adds recursive smoothing of the finite-difference scalar at the same cost. We show that the ambient finite difference agrees with its retracted counterpart to leading order, that the normalized estimator recovers the Riemannian gradient in expectation up to an $\mathcal{O}(\epsilon^2)$ bias, and that ZO-RGD admits a nonasymptotic convergence guarantee conditional on trajectory regularity. Both variants are competitive with strong forward-only baselines on the OPT suite, under a controlled symmetric protocol, and on Llama-3.1-8B and Qwen2.5-7B, with gains in several settings that mechanism tests attribute to tangent normalization and the manifold parameterization.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing the methods, proving the conditional convergence result, and to improve the readability of the paper. We have not used generative AI tools for any other task with required disclosure. We have reviewed all AI-assisted work: the code was tested by the authors, and all text edits were checked to preserve the technical content. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper proposes a memory-efficient, forward-only optimizer for fine-tuning large language models and evaluates it on publicly available pretrained models (OPT, Llama-3.1, and Qwen2.5) and public benchmark data sets. The work involves no human subjects, no newly collected data, and no private or personally identifiable information. Lowering the memory required for fine-tuning broadens access to model adaptation; this carries the dual-use considerations common to all fine-tuning methods, since adapted models can serve beneficial or harmful ends, but our method introduces no risk beyond that of existing fine-tuning approaches.

REPRODUCIBILITY STATEMENT

The proposed methods are fully specified: ZO-RGD in Algorithm 1 (Section 4.3) and ZO-RGD-S in Algorithm 2 (Appendix A.3), with the stored representation, initialization, retraction, and per-step cost detailed in Appendix A. All assumptions underlying the theoretical results are stated in Appendix C.2; complete proofs of the propositions and of the convergence theorem are given in Appendices B and C, respectively. The evaluation protocol is described in Section 6.1 and Appendix D.1, compute resources in Appendix D.2, and per-seed values in Appendix D.7. All experiments use publicly available pretrained models and public data sets. The source code will be released upon acceptance, together with a manifest that maps every reported cell to its job configuration, selection rule, seeds, hardware, and completion status, including failed runs.

REFERENCES

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 9780691132983. doi: 10.1515/9781400830244. URL https://press.princeton.edu/books/hardcover/9780691132983/optimization-algorithms-on-matrix-manifolds.

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.

Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points. *Foundations of Computational Mathematics*, 22(1):35–76, 2022. doi: 10.1007/s10208-021-09499-8. URL https://doi.org/10.1007/s10208-021-09499-8.

Albert S. Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, 22(2):507–560, 2022. doi: 10.1007/s10208-021-09513-z. URL https://doi.org/10.1007/s10208-021-09513-z.

Fengmiao Bian, Jinyang Zheng, Ziyun Liu, Jianzhou Luo, and Jian-Feng Cai. Finding low-rank matrix weights in DNNs via Riemannian optimization: RAdaGrad and RAdamW. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tiGFiCrmKm.

Vladimir Bogachev, Vladimir Aletov, Alexander Molozhavenko, Denis Bobkov, Vera Soboleva, Aibek Alanov, and Maxim Rakhuba. LoRA meets Riemannion: Muon optimizer for parametrization-independent low-rank adapters. arXiv:2507.12142, July 2025. URL https://arxiv.org/abs/2507.12142.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Konstantinos Parasyris, Jiancheng Liu, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. DeepZero: Scaling up zeroth-order optimization for deep model training. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=qBWhjsNPEY.

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

Sizhe Dang, Haishan Ye, Yangyang Guo, Xiaodong Zheng, Yanjun Zhao, Guang Dai, et al. FZOO: Fast zeroth-order optimizer for fine-tuning large language models towards Adam-scale speed. *arXiv preprint arXiv:2506.09034*, 2025.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246/.

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 Trans. Inf. Theory*, 61(5):2788–2806, Mar. 2015. doi: 10.1109/TIT.2015.2409256. URL https://doi.org/10.1109/TIT.2015.2409256.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. doi: 10.1137/120880811. URL https://doi.org/10.1137/120880811.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17783–17806. PMLR, July 2024. URL https://proceedings.mlr.press/v235/hayou24a.html.

Chang He, Zhaoye Pan, Xiao Wang, and Bo Jiang. Riemannian accelerated zeroth-order algorithm: Improved robustness and lower query complexity. *arXiv preprint arXiv:2405.05713*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. HiRA: Parameter-efficient Hadamard high-rank adaptation for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=TwJrTz9cRS.

Jiaxiang Li, Krishnakumar Balasubramanian, and Shiqian Ma. Stochastic zeroth-order Riemannian derivative estimation and optimization. *Mathematics of Operations Research*, 48(2):1183–1211, 2022. doi: 10.1287/moor.2022.1302. URL https://pubsonline.informs.org/doi/abs/10.1287/moor.2022.1302.

Jiaxiang Li, Krishnakumar Balasubramanian, and Shiqian Ma. Zeroth-order riemannian averaging stochastic approximation algorithms. *arXiv preprint arXiv:2309.14506*, 2023.

Shiwei Li, Xiandi Luo, Xing Tang, Haozhao Wang, Hao Chen, Weihong Luo, Yuhua Li, Xiuqiang He, and Ruixuan Li. Beyond zero initialization: Investigating the impact of non-zero initialization on LoRA fine-tuning dynamics. In *Proceedings of the 42nd International Conference on Machine*

Learning, volume 267 of *Proceedings of Machine Learning Research*, pp. 35519–35535. PMLR, July 2025. URL https://proceedings.mlr.press/v267/li25bm.html.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32100–32121. PMLR, July 2024a. URL https://proceedings.mlr.press/v235/liu24bn.html.

Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Sparse MeZO: Less parameters for better performance in zeroth-order LLM fine-tuning. *arXiv preprint arXiv:2402.15751*, 2024b.

Shaocong Ma and Heng Huang. Riemannian zeroth-order gradient estimation with structure-preserving metrics for geodesically incomplete manifolds. In *The Fourteenth International Conference on Learning Representations*, 2026.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53038–53075, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a627810151be4d13f907ac898ff7e948-Abstract-Conference.html.

Zhendong Mi, Qitao Tan, Grace Li Zhang, Zhaozhuo Xu, Geng Yuan, and Shaoyi Huang. Towards fast LLM fine-tuning through zeroth-order optimization with projected gradient-aligned perturbations. arXiv:2510.18228, Oct. 2025. URL https://arxiv.org/abs/2510.18228.

Zhanfeng Mo, Long-Kai Huang, and Sinno Jialin Pan. Parameter and memory efficient pretraining via low-rank Riemannian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=i0zzO7Hslk.

Yurii Nesterov and Vladimir Spokoiny. Random Gradient-Free Minimization of Convex Functions. *Found. Comput. Math.*, 17(2):527–566, Apr. 2017. ISSN 1615-3383. doi: 10.1007/s10208-015-9296-2. URL https://doi.org/10.1007/s10208-015-9296-2.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Reinhold Schneider and André Uschmajew. Convergence results for projected line-search methods on varieties of low-rank matrices via łojasiewicz inequality. *SIAM Journal on Optimization*, 25(1):622–646, 2015. doi: 10.1137/140957822. URL https://doi.org/10.1137/140957822.

Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18(52):1–11, 2017. URL https://jmlr.org/papers/v18/16-632.html.

Hao Nan Sheng, Zhi-Yong Wang, Mingrui Yang, and Hing Cheung So. AROMA: Autonomous rank-one matrix adaptation. arXiv:2504.05343, Apr. 2025. URL https://arxiv.org/abs/2504.05343.

Zichen Song and Weijia Li. RoZO: Geometry-aware zeroth-order fine-tuning on low-rank adapters for black-box large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1773–1783, Rabat, Morocco, Mar. 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.eacl-long.80. URL https://aclanthology.org/2026.eacl-long.80/.

James C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, Mar. 1992. doi: 10.1109/9.119632. URL https://doi.org/10.1109/9.119632.

Bart Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013. doi: 10.1137/110845768. URL https://doi.org/10.1137/110845768.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Shuo Wang, Ziyu Chen, and Ming Tang. CurvZO: Adaptive curvature-guided sparse zeroth-order optimization for efficient LLM fine-tuning. arXiv:2603.21725, Mar. 2026. URL https://arxiv.org/abs/2603.21725.

Ke Wei, Jian-Feng Cai, Tony F. Chan, and Shingyu Leung. Guarantees of Riemannian optimization for low rank matrix recovery. *SIAM Journal on Matrix Analysis and Applications*, 37 (3):1198–1222, 2016. doi: 10.1137/15M1050525. URL https://doi.org/10.1137/15M1050525.

Menglin Yang, Jialin Chen, Jinkai Tao, Yifei Zhang, Jiahong Liu, Jiasheng Zhang, Qiyao Ma, Harshit Verma, Regina Zhang, Min Zhou, Irwin King, and Rex Ying. Low-rank adaptation for foundation models: A comprehensive review. arXiv:2501.00365, Dec. 2024a. URL https://arxiv.org/abs/2501.00365.

Yifan Yang, Kai Zhen, Ershad Banijamali, Athanasios Mouchtaris, and Zheng Zhang. AdaZeta: Adaptive zeroth-order tensor-train adaption for memory-efficient large language models fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 977–995, Miami, Florida, USA, Nov. 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.56. URL https://aclanthology.org/2024.emnlp-main.56/.

Yifan Yang, Jiajun Zhou, Ngai Wong, and Zheng Zhang. LoRETTA: Low-rank economic tensor-train adaptation for ultra-low-parameter fine-tuning of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3161–3176, Mexico City, Mexico, June 2024c. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.174. URL https://aclanthology.org/2024.naacl-long.174/.

Can Yaras, Peng Wang, Laura Balzano, and Qing Qu. Compressible dynamics in deep overparameterized low-rank learning and adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56946–56965. PMLR, July 2024. URL https://proceedings.mlr.press/v235/yaras24a.html.

Ziming Yu, Pan Zhou, Sike Wang, Jia Li, Mi Tian, and Hua Huang. Zeroth-order fine-tuning of LLMs in random subspaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4475–4485, Oct. 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Yu_Zeroth-Order_Fine-Tuning_of_LLMs_in_Random_Subspaces_ICCV_2025_paper.html.

Fangzhao Zhang and Mert Pilanci. Riemannian preconditioned LoRA for fine-tuning foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 59641–59669. PMLR, July 2024. URL https://proceedings.mlr.press/v235/zhang24ax.html.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer,

Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. arXiv:2205.01068, May 2022. URL https://arxiv.org/abs/2205.01068.

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D. Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 59173–59190. PMLR, July 2024. URL https://proceedings.mlr.press/v235/zhang24ad.html.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61121–61143. PMLR, July 2024a. URL https://proceedings.mlr.press/v235/zhao24s.html.

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:2402.15173, Feb. 2024b. URL https://arxiv.org/abs/2402.15173.

A Algorithmic and Implementation Details

This appendix follows the order of Section 4: the thin-SVD representation (Appendix A.1), the near-zero initialization of the fixed-rank state (Appendix A.2), the full pseudocode of ZO-RGD-S (Appendix A.3), and the complexity of the implementation (Appendix A.4).

A.1 Thin-SVD representation

The optimization variable is the matrix $\Delta \boldsymbol{W}_\ell$ itself, and the stored thin-SVD triple $(\boldsymbol{U}_\ell, \boldsymbol{\Sigma}_\ell, \boldsymbol{V}_\ell)$ is a numerical representation of that point. The representation is unique only up to simultaneous sign changes of paired singular vectors, and, when singular values coincide, up to joint rotations of the corresponding subspaces. None of this affects the algorithm or the analysis, because projection, normalization, and retraction depend on the represented point only: they are invariant to the choice of representative. This is the precise sense in which the method has no factor-gauge degrees of freedom, in contrast to optimizing LoRA factor coordinates $(\boldsymbol{B}, \boldsymbol{A})$.

A.2 Near-zero initialization

The fixed-rank constraint excludes the zero update when $r_\ell > 0$. To preserve the pretrained initialization while remaining feasible for ZO-RGD and ZO-RGD-S, we initialize each trainable update block at a near-zero fixed-rank point:

$$\Delta \boldsymbol{W}_{\mathrm{init},\ell} = \boldsymbol{U}_{\mathrm{init},\ell}(\sigma_{\mathrm{init},\ell} \boldsymbol{I}_{r_\ell}) \boldsymbol{V}_{\mathrm{init},\ell}^\top, \qquad \ell \in \mathcal{S}, \tag{19}$$

where $0 < \sigma_{\mathrm{init},\ell} \ll 1$, and $\boldsymbol{U}_{\mathrm{init},\ell}$ and $\boldsymbol{V}_{\mathrm{init},\ell}$ have orthonormal columns. The factors are obtained by QR orthonormalization of Gaussian matrices (computed in single precision and cast to the model precision), and $\sigma_{\mathrm{init},\ell} = 10^{-6}$ by default. This keeps each update block on its prescribed fixed-rank manifold while making $\boldsymbol{W}_0 + \Delta \boldsymbol{W}_{\mathrm{init}}$ numerically close to the pretrained model.

A.3 ZO-RGD-S pseudocode

Algorithm 2 expands the scalar recursion in (14). As in Algorithm 1, the pseudocode shows the matrix-block updates.

Algorithm 2 ZO-RGD-S

Require: Frozen weights $\boldsymbol{W}_0$; initial update $\Delta\boldsymbol{W}_0 \in \mathcal{M}$; learning rate η; perturbation radius ϵ; smoothing parameter $\beta \in [0, 1)$.

Fix an order $\ell_1, \ldots, \ell_K$ of the trainable matrix blocks.
for $t = 0, 1, 2, \ldots$ **do**
  Sample a realization ξ_t, construct normalized tangent random matrices, and compute $\hat{s}_t$ with the same two forward passes as ZO-RGD.
  if $t = 0$ **then**
    $m_{0,0} = \hat{s}_0$.
  else
    $m_{t,0} = m_{t-1,K}$.
  end if
  for $k = 1, \ldots, K$ **do**
    $m_{t,k} = \beta m_{t,k-1} + (1 - \beta)\hat{s}_t$.
    $\widehat{\boldsymbol{G}}^{\mathrm{S}}_{t,\ell_k} = m_{t,k}\widetilde{\boldsymbol{Z}}_{t,\ell_k}$.
    $\Delta\boldsymbol{W}_{t+1,\ell_k} = \mathcal{R}_{\Delta\boldsymbol{W}_{t,\ell_k}}(-\eta\widehat{\boldsymbol{G}}^{\mathrm{S}}_{t,\ell_k})$.
  end for
end for

A.4 Complexity of the implementation

Two costs are distinguished: the persistent trainable state and the per-step work.

Persistent state. At each selected two-dimensional block, the trainable update is stored as the thin-SVD triple $(\boldsymbol{U}_\ell, \boldsymbol{\Sigma}_\ell, \boldsymbol{V}_\ell)$, with $\boldsymbol{U}_\ell \in \mathbb{R}^{m_\ell \times r_\ell}$, diagonal $\boldsymbol{\Sigma}_\ell \in \mathbb{R}^{r_\ell \times r_\ell}$, and $\boldsymbol{V}_\ell \in \mathbb{R}^{n_\ell \times r_\ell}$. The per-block state therefore has $r_\ell(m_\ell + n_\ell + 1)$ parameters, i.e. $O((m_\ell + n_\ell)r_\ell)$, much smaller than a dense $m_\ell n_\ell$ trainable update when $r_\ell \ll \min(m_\ell, n_\ell)$. The training loop stores no first-order optimizer state and no backpropagation graph; random matrices are regenerated from seeds, and the thin-SVD triple is the only trainable representation. ZO-RGD-S adds only a scalar smoothing variable and does not maintain a matrix-valued momentum buffer. Thus the forward-only memory profile remains in the MeZO/LOZO regime, with an additional fixed-rank update state that scales linearly in rank; Appendix D.17 quantifies this state for OPT-1.3B.

Per-step work. The per-step work, however, is not low-rank: the current implementation materializes dense $m_\ell \times n_\ell$ intermediates, the Gaussian random matrix, its tangent projection, and the update direction, so each block incurs $O(m_\ell n_\ell r_\ell)$ work in the projection and retraction products plus $O(m_\ell n_\ell)$ transient memory, in addition to the two forward passes; only the core SVD of the retraction is small ($2r_\ell \times 2r_\ell$). In two allocations on different A100-80G chips, profiling all methods back to back gives roughly $1.8\times$ the per-step time of LOZO at the identical forward-pass count (Table 6). Projection and retraction add computation while retaining comparable peak memory. A fused factored application that samples random matrices directly in factor coordinates could avoid the dense intermediates in principle, but it is an engineering change we have not implemented or measured, and a "lazy" retraction applied every few steps would change the algorithm itself (intermediate iterates would leave the manifold and random matrices would use a stale tangent space), requiring its own bias and convergence analysis.

B Proofs for Tangent-Normalized ZO Estimation

This appendix proves Propositions 1 and 2 of Section 5. For the reader's convenience, each proposition is restated under its original number immediately before its proof.

Proposition 1 (Ambient and retracted finite differences)**.** *Let $F(\cdot; \xi_t)$ be differentiable near $\Delta\boldsymbol{W}_t \in \mathcal{M}_r$, and let $\mathcal{R}_{\Delta\boldsymbol{W}_t}(\tau\boldsymbol{Y}) = \Delta\boldsymbol{W}_t + \tau\boldsymbol{Y} + \mathcal{O}(\tau^2)$ as $\tau \to 0$ for every tangent $\boldsymbol{Y}$. If $\hat{s}_t^{\mathrm{retr}}$ denotes (11) evaluated at $\mathcal{R}_{\Delta\boldsymbol{W}_t}(\pm\epsilon\widetilde{\boldsymbol{Z}}_t)$, then $\hat{s}_t = \hat{s}_t^{\mathrm{retr}} + o(1)$ as $\epsilon \to 0$.*

Write the implemented scalar $\hat{s}_t$ of (11) as $\hat{s}_t^{\text{lin}}$, and its retracted counterpart as $\hat{s}_t^{\text{retr}}$:

$$\hat{s}_t^{\text{lin}} = \frac{F(\Delta \boldsymbol{W}_t + \epsilon \widetilde{\boldsymbol{Z}}_t; \xi_t) - F(\Delta \boldsymbol{W}_t - \epsilon \widetilde{\boldsymbol{Z}}_t; \xi_t)}{2\epsilon}, \tag{20}$$

$$\hat{s}_t^{\text{retr}} = \frac{F(\mathcal{R}_{\Delta \boldsymbol{W}_t}(\epsilon \widetilde{\boldsymbol{Z}}_t); \xi_t) - F(\mathcal{R}_{\Delta \boldsymbol{W}_t}(-\epsilon \widetilde{\boldsymbol{Z}}_t); \xi_t)}{2\epsilon}. \tag{21}$$

Proof. The linearized finite difference satisfies

$$\hat{s}_t^{\text{lin}} = \left\langle \nabla_{\Delta \boldsymbol{W}} F(\Delta \boldsymbol{W}_t; \xi_t), \widetilde{\boldsymbol{Z}}_t \right\rangle_{\text{F}} + o(1).$$

By the retraction property,

$$\mathcal{R}_{\Delta \boldsymbol{W}_t}(\pm \epsilon \widetilde{\boldsymbol{Z}}_t) = \Delta \boldsymbol{W}_t \pm \epsilon \widetilde{\boldsymbol{Z}}_t + \mathcal{O}(\epsilon^2).$$

Substituting this expansion into the retracted finite difference gives the same leading directional derivative. Hence

$$\hat{s}_t^{\text{retr}} = \left\langle \nabla_{\Delta \boldsymbol{W}} F(\Delta \boldsymbol{W}_t; \xi_t), \widetilde{\boldsymbol{Z}}_t \right\rangle_{\text{F}} + o(1),$$

which proves the claim. □

Proposition 2 (Expected gradient alignment)**.** *Fix $\Delta \boldsymbol{W}_t \in \mathcal{M}_r$ and ξ_t, and assume $F(\cdot; \xi_t)$ has a locally Lipschitz Hessian near $\Delta \boldsymbol{W}_t$. For the estimator in (12) and the Frobenius metric, as $\epsilon \to 0$,*

$$\mathbb{E}_{\boldsymbol{Z}_t}[\widehat{\boldsymbol{G}}_t \mid \Delta \boldsymbol{W}_t, \xi_t] = \frac{mn}{d_r} \operatorname{grad}_{\mathcal{M}_r} F(\Delta \boldsymbol{W}_t; \xi_t) + \mathcal{O}(\epsilon^2).$$

If $\mathbb{E}_{\xi_t}[F] = f$ and differentiation commutes with expectation, the same holds with f in place of F.

Proof. Let

$$T_t := T_{\Delta \boldsymbol{W}_t} \mathcal{M}_r, \qquad d_r = \dim(T_t) = r(m + n - r).$$

Since $\widehat{\boldsymbol{Z}}_t = P_{T_t}(\boldsymbol{Z}_t)$ is the projection of an isotropic Gaussian matrix onto the tangent space, the normalized vector

$$\boldsymbol{u}_t := \frac{\widehat{\boldsymbol{Z}}_t}{\|\widehat{\boldsymbol{Z}}_t\|_{\text{F}}}$$

is isotropic on the unit sphere of T_t. Thus

$$\widetilde{\boldsymbol{Z}}_t = \sqrt{mn}\, \boldsymbol{u}_t.$$

Assuming that $F(\cdot; \xi_t)$ is sufficiently smooth in a neighborhood of $\Delta \boldsymbol{W}_t$, a Taylor expansion for fixed ξ_t gives

$$\hat{s}_t = \frac{F(\Delta \boldsymbol{W}_t + \epsilon \widetilde{\boldsymbol{Z}}_t; \xi_t) - F(\Delta \boldsymbol{W}_t - \epsilon \widetilde{\boldsymbol{Z}}_t; \xi_t)}{2\epsilon} \tag{22}$$

$$= \left\langle \nabla_{\Delta \boldsymbol{W}} F(\Delta \boldsymbol{W}_t; \xi_t), \widetilde{\boldsymbol{Z}}_t \right\rangle_{\text{F}} + \mathcal{O}(\epsilon^2). \tag{23}$$

Let

$$\boldsymbol{g}_t := \operatorname{grad}_{\mathcal{M}_r} F(\Delta \boldsymbol{W}_t; \xi_t) = P_{T_t}\left(\nabla_{\Delta \boldsymbol{W}} F(\Delta \boldsymbol{W}_t; \xi_t)\right)$$

be the Riemannian gradient under the Frobenius metric. Since $\boldsymbol{u}_t \in T_t$, we have

$$\langle \nabla_{\Delta \boldsymbol{W}} F(\Delta \boldsymbol{W}_t; \xi_t), \boldsymbol{u}_t \rangle_{\text{F}} = \langle \boldsymbol{g}_t, \boldsymbol{u}_t \rangle_{\text{F}}.$$

Therefore,

$$\widehat{\boldsymbol{G}}_t = \hat{s}_t \widetilde{\boldsymbol{Z}}_t = mn\, \langle \boldsymbol{g}_t, \boldsymbol{u}_t \rangle_{\text{F}}\, \boldsymbol{u}_t + \mathcal{O}(\epsilon^2), \tag{24}$$

where the remainder is in the Frobenius norm. By isotropy on the d_r-dimensional tangent space,

$$\mathbb{E}_{\boldsymbol{Z}_t}\left[\langle \boldsymbol{g}_t, \boldsymbol{u}_t \rangle_{\text{F}}\, \boldsymbol{u}_t \mid \Delta \boldsymbol{W}_t, \xi_t\right] = \frac{1}{d_r} \boldsymbol{g}_t.$$

Hence

$$\mathbb{E}_{\boldsymbol{Z}_t}\left[\widehat{\boldsymbol{G}}_t \mid \Delta \boldsymbol{W}_t, \xi_t\right] = \frac{mn}{d_r} \operatorname{grad}_{\mathcal{M}_r} F(\Delta \boldsymbol{W}_t; \xi_t) + \mathcal{O}(\epsilon^2).$$

If $\mathbb{E}_{\xi_t}[F(\Delta \boldsymbol{W}; \xi_t)] = f(\Delta \boldsymbol{W})$ and differentiation commutes with expectation, then taking expectation over ξ_t gives the second statement. □

C CONVERGENCE ANALYSIS OF ZO-RGD

This appendix proves the results of Section 5 for ZO-RGD exactly as implemented: the mixed product of fixed-rank blocks and a Euclidean factor of one-dimensional parameters, blockwise normalized tangent random matrices, one shared finite-difference scalar per step, ambient tangent-line evaluation points, and update-only truncated-SVD retraction. Proposition 2 is the single-block specialization of Lemmas 1–3. Throughout, "ZO-RGD" means the base method of Algorithm 1 with one joint random direction per step, zero weight decay, no singular-value clamping, no gradient accumulation, and a differentiable training loss.

C.1 ALGORITHM RESTATEMENT

Blocks and constants. The trainable two-dimensional blocks are indexed by $\ell \in \{1, \dots, N\}$; block ℓ has ambient shape $m_\ell \times n_\ell$ and fixed rank $1 \le r_\ell \le \min(m_\ell, n_\ell)$. (Throughout, N is the block count; the symbol L is reserved for the gradient-Lipschitz constant of Assumption 2, and ρ_3 for its third-derivative constant, distinct from the normalization radii ρ_ℓ.) Write $\mathcal{M}_\ell := \mathcal{M}_{r_\ell}^{m_\ell \times n_\ell}$ for the fixed-rank manifold of block ℓ,

$$d_\ell := \dim \mathcal{M}_\ell = r_\ell(m_\ell + n_\ell - r_\ell), \qquad \rho_\ell^2 := m_\ell n_\ell, \qquad c_\ell := \frac{\rho_\ell^2}{d_\ell}.$$

The trainable one-dimensional parameters (biases and LayerNorm parameters) form one Euclidean factor $\mathbb{R}^{d_0}$; set $c_0 := 1$. $c_\ell \ge 1$, because $m_\ell n_\ell - d_\ell = (m_\ell - r_\ell)(n_\ell - r_\ell) \ge 0$, with equality if and only if $r_\ell = \min(m_\ell, n_\ell)$.

State and dimensions. The trainable state is

$$\boldsymbol{x} = (\Delta \boldsymbol{W}_1, \dots, \Delta \boldsymbol{W}_N, \boldsymbol{b}) \in \mathcal{X} := \left(\prod_{\ell=1}^N \mathcal{M}_\ell\right) \times \mathbb{R}^{d_0},$$

embedded in the ambient Euclidean product $\mathcal{E}_{\mathrm{mix}} := \left(\prod_\ell \mathbb{R}^{m_\ell \times n_\ell}\right) \times \mathbb{R}^{d_0}$ with the direct-sum Frobenius inner product. The total ambient trainable dimension is $\bar{D} := \sum_\ell m_\ell n_\ell + d_0$.

Loss function. $F(\boldsymbol{x}; \xi) := \mathcal{L}(\boldsymbol{W}_0 + \Delta \boldsymbol{W}, \boldsymbol{b}; \xi)$, defined on an open neighborhood of $\mathcal{X}$ in $\mathcal{E}_{\mathrm{mix}}$: the finite-difference points below leave $\mathcal{X}$, so F must be (and, in the implementation, is) an ambient function. Set $f(\boldsymbol{x}) := \mathbb{E}_\xi F(\boldsymbol{x}; \xi)$.

Tangent spaces and gradients. $T_{\boldsymbol{x}} := \left(\prod_\ell T_\ell\right) \times \mathbb{R}^{d_0}$ with $T_\ell := T_{\Delta \boldsymbol{W}_\ell} \mathcal{M}_\ell$, and P_ℓ denotes the orthogonal projector onto T_ℓ from (4). The Riemannian gradient in the product Frobenius metric is $\operatorname{grad} f(\boldsymbol{x}) = (P_1 \nabla_1 f, \dots, P_N \nabla_N f, \nabla_0 f)$, where ∇_ℓ (∇_0) is the Euclidean partial gradient with respect to block ℓ (the 1D factor).

Weighted norm. Define the linear map $\boldsymbol{D} := \operatorname{blkdiag}(c_1 I_{T_1}, \dots, c_N I_{T_N}, I_{d_0})$ on $T_{\boldsymbol{x}}$ and, for $\boldsymbol{v} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_N, \boldsymbol{v}_0) \in T_{\boldsymbol{x}}$, the weighted norm

$$\|\boldsymbol{v}\|_{\boldsymbol{D}}^2 := \sum_\ell c_\ell \|\boldsymbol{v}_\ell\|_{\mathrm{F}}^2 + \|\boldsymbol{v}_0\|^2.$$

Since every $c_\ell \ge 1$ and $c_0 = 1$, we have $\|\boldsymbol{v}\|_{\boldsymbol{D}}^2 \ge \|\boldsymbol{v}\|^2$ (plain product norm).

At step t, given $\boldsymbol{x}_t = (\Delta \boldsymbol{W}_{1,t}, \dots, \Delta \boldsymbol{W}_{L,t}, \boldsymbol{b}_t)$, we restate the proposed algorithm as follow:

1. Draw $\boldsymbol{Z}_\ell \sim \mathcal{N}(0, I_{m_\ell \times n_\ell})$ i.i.d. across ℓ and $\boldsymbol{z}_0 \sim \mathcal{N}(0, I_{d_0})$, independent of the past and of ξ_t.
2. Set $\widehat{\boldsymbol{Z}}_\ell := P_\ell \boldsymbol{Z}_\ell$ and $\widetilde{\boldsymbol{Z}}_\ell := \rho_\ell \, \widehat{\boldsymbol{Z}}_\ell / \|\widehat{\boldsymbol{Z}}_\ell\|_{\mathrm{F}}$ (almost surely well defined). The joint random direction is $\boldsymbol{u}_t := (\widetilde{\boldsymbol{Z}}_1, \dots, \widetilde{\boldsymbol{Z}}_N, \boldsymbol{z}_0) \in T_{\boldsymbol{x}_t}$.
3. Two function evaluations on the ambient line, sharing the same minibatch ξ_t: $\hat{s}_t := \big[F(\boldsymbol{x}_t + \epsilon \boldsymbol{u}_t; \xi_t) - F(\boldsymbol{x}_t - \epsilon \boldsymbol{u}_t; \xi_t)\big]/(2\epsilon)$.
4. Estimator $\widehat{\boldsymbol{G}}_t := \hat{s}_t \, \boldsymbol{u}_t$, so $\widehat{\boldsymbol{G}}_{t,\ell} = \hat{s}_t \widetilde{\boldsymbol{Z}}_\ell \in T_\ell$ and $\widehat{\boldsymbol{G}}_{t,0} = \hat{s}_t \boldsymbol{z}_0$.

5. Update: $\Delta \boldsymbol{W}_{\ell,t+1} = \mathcal{H}_{r_\ell}(\Delta \boldsymbol{W}_{\ell,t} - \eta \widehat{\boldsymbol{G}}_{t,\ell})$ for each block and $\boldsymbol{b}_{t+1} = \boldsymbol{b}_t - \eta \widehat{\boldsymbol{G}}_{t,0}$; that is, $\boldsymbol{x}_{t+1} = \mathcal{R}_{\boldsymbol{x}_t}(-\eta \widehat{\boldsymbol{G}}_t)$ with the product retraction $\mathcal{R} = (\mathcal{H}_{r_1}, \ldots, \mathcal{H}_{r_L}, \mathrm{id})$.

Remark 2 (randomness model). *Let $\mathcal{F}_t := \sigma(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t)$; the draws $(\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_N, \boldsymbol{z}_0, \xi_t)$ are independent of $\mathcal{F}_t$. Three standard idealizations bridge the implementation and the probabilistic model. (i) Per-step seeds come from one pseudorandom stream; we model them as i.i.d. (ii) Assumption 3 treats minibatches as conditionally unbiased given $\mathcal{F}_t$, whereas the data loader shuffles a finite dataset and samples without replacement within an epoch; this is the standard stochastic-oracle idealization shared with the MeZO and LOZO analyses. (iii) The common-random-numbers cancellation in Lemma 2 requires the forward pass to be deterministic given $(\boldsymbol{x}, \xi)$, which holds in the implementation (evaluation mode; no dropout). Conditional expectations below are justified by the finite second moment (Lemma 4) and the $\mathcal{F}_t$-measurability of* $\operatorname{grad} f(\boldsymbol{x}_t)$.

C.2 Assumptions and technical lemmas

Assumption 1 (trajectory regularity). *There exist $\sigma_{\min} > 0$, $G_\infty < \infty$, and $f_* > -\infty$ such that, almost surely along the realized trajectory, finite-difference segments, and update segments: (i) $\sigma_{r_\ell}(\Delta \boldsymbol{W}_{\ell,t}) \geq \sigma_{\min}$ for every block and step; (ii) $\|\nabla f\| \leq G_\infty$; (iii) $f \geq f_*$; (iv) $\eta \|\widehat{\boldsymbol{G}}_{t,\ell}\|_{\mathrm{F}} \leq \sigma_{\min}/2$ for every step t and 2D block ℓ.*

Assumption 2 (smoothness). *$F(\cdot; \xi)$ is differentiable with L-Lipschitz gradient, uniformly in ξ, and three times differentiable along the realized finite-difference segments with third directional derivatives bounded: $|\phi'''(\tau)| \leq \rho_3 \|\boldsymbol{v}\|^3$ for $\phi(\tau) := F(\boldsymbol{x} + \tau \boldsymbol{v}; \xi)$.*

Assumption 3 (stochastic model). *$\mathbb{E}_\xi \nabla F(\boldsymbol{x}; \xi) = \nabla f(\boldsymbol{x})$, with blockwise variances $\mathbb{E}_\xi \|P_\ell(\nabla_\ell F - \nabla_\ell f)\|_{\mathrm{F}}^2 \leq \sigma_\ell^2$ and $\mathbb{E}_\xi \|\nabla_0 F - \nabla_0 f\|^2 \leq \sigma_0^2$; both function evaluations of step t share the same minibatch ξ_t.*

Assumptions 1–3 in Section 5 are used with the following precise domain. Let $\mathcal{X}_{\mathrm{reg}} \subset \mathcal{X}$ be a set containing all iterates almost surely, and define its enlargement

$$\mathcal{X}'_{\mathrm{reg}} := \{\boldsymbol{x}_t + \tau \boldsymbol{u}_t : |\tau| \leq \epsilon\} \cup \{\boldsymbol{x}_t - \tau \eta \widehat{\boldsymbol{G}}_t : \tau \in [0,1]\} \cup \{\text{segments between } \boldsymbol{x}_t - \eta \widehat{\boldsymbol{G}}_t \text{ and } \mathcal{R}_{\boldsymbol{x}_t}(-\eta \widehat{\boldsymbol{G}}_t)\}, \tag{25}$$

over all steps t and realized draws. Assumption 1 requires: (i) $\sigma_{r_\ell}(\Delta \boldsymbol{W}_{\ell,t}) \geq \sigma_{\min} > 0$ for every block and step on $\mathcal{X}_{\mathrm{reg}}$; (ii) $\|\nabla f\| \leq G_\infty$ on $\mathcal{X}'_{\mathrm{reg}}$; (iii) $f \geq f_* > -\infty$; (iv) almost surely, $\eta \|\widehat{\boldsymbol{G}}_{t,\ell}\|_{\mathrm{F}} \leq \sigma_{\min}/2$ for every t and every 2D block ℓ. Assumption 2 requires $F(\cdot; \xi)$ differentiable with L-Lipschitz gradient on $\mathcal{X}'_{\mathrm{reg}}$, uniformly in ξ, and, along every realized finite-difference segment, $\phi(\tau) := F(\boldsymbol{x} + \tau \boldsymbol{v}; \xi)$ three times differentiable with $|\phi'''(\tau)| \leq \rho_3 \|\boldsymbol{v}\|^3$. Assumption 3 requires $\mathbb{E}_\xi \nabla F(\boldsymbol{x}; \xi) = \nabla f(\boldsymbol{x})$ on $\mathcal{X}'_{\mathrm{reg}}$ and the blockwise variance bounds $\mathbb{E}_\xi \|P_\ell(\nabla_\ell F - \nabla_\ell f)\|_{\mathrm{F}}^2 \leq \sigma_\ell^2$ and $\mathbb{E}_\xi \|\nabla_0 F - \nabla_0 f\|^2 \leq \sigma_0^2$ on $\mathcal{X}_{\mathrm{reg}}$, with both function evaluations sharing the same ξ_t. As a consequence of Assumptions 2 and 3, $f = \mathbb{E}_\xi F$ is differentiable with L-Lipschitz gradient on $\mathcal{X}'_{\mathrm{reg}}$ (dominated convergence plus uniformity in ξ); this is used in Lemma 6. Write

$$\sigma_D^2 := \sum_\ell c_\ell \sigma_\ell^2 + \sigma_0^2, \qquad \Delta_0 := f(\boldsymbol{x}_0) - f_*.$$

Remark 3 (what Assumption 1 does and does not assert). *(a)* Regular region. *Each $\mathcal{M}_\ell$ is non-compact: its closure adds the rank-deficient boundary, on which the tangent spaces and the retraction degenerate. Assumption 1 confines the analysis to a region away from that boundary; it does* not *prove that the iterates stay there. This is the standard form of assumption in fixed-rank optimization (Vandereycken, 2013; Schneider & Uschmajew, 2015). Clause (i) is checkable because the state stores $\boldsymbol{\Sigma}_\ell$ explicitly; since the near-zero initialization (19) starts at distance $\sigma_{\mathrm{init},\ell}$ from the boundary, the clause is most demanding early in training, and Appendix D.15 checks it empirically on our runs. (b)* Step regularity. *Clause (iv) is not enforced by the algorithm. A deterministic sufficient condition: if $|F| \leq B_F$ on $\mathcal{X}'_{\mathrm{reg}}$, then $|\hat{s}_t| \leq B_F/\epsilon$ and $\|\widetilde{\boldsymbol{Z}}_\ell\|_{\mathrm{F}} = \rho_\ell$, so $\eta \leq \epsilon\, \sigma_{\min}/(2 B_F \max_\ell \rho_\ell)$ implies (iv). (c)* 1D segments. *On the Euclidean factor, $\|\widehat{\boldsymbol{G}}_{t,0}\| = |\hat{s}_t| \|\boldsymbol{z}_0\|$ has no deterministic bound, so uniform constants (L, ρ_3, G_∞) on $\mathcal{X}'_{\mathrm{reg}}$ are a nontrivial requirement in the 1D directions. Alternatively, one can condition on the event $\|\boldsymbol{z}_0\| \leq \sqrt{d_0} + \sqrt{2 \log(T/\delta)}$, of probability at least $1 - \delta$, and absorb its complement into an additive δ-term; we state the assumption version for cleanliness.*

Lemma 1 (random-direction law and moments). *Condition on $\mathcal{F}_t$ (equivalently, fix the tangent spaces T_ℓ; note T_ℓ is a random, $\mathcal{F}_t$-measurable subspace — only its dimension d_ℓ is deterministic, since Assumption 1(i) keeps every iterate at exact rank r_ℓ). Then $\widehat{\boldsymbol{Z}}_\ell = P_\ell \boldsymbol{Z}_\ell$ is a standard Gaussian on the d_ℓ-dimensional subspace T_ℓ, $\boldsymbol{u}_\ell := \widehat{\boldsymbol{Z}}_\ell / \|\widehat{\boldsymbol{Z}}_\ell\|_{\mathrm{F}}$ is uniform on the unit sphere of T_ℓ, and $\widetilde{\boldsymbol{Z}}_\ell = \rho_\ell \boldsymbol{u}_\ell$ is uniform on the sphere of radius ρ_ℓ in T_ℓ; in particular $\|\widetilde{\boldsymbol{Z}}_\ell\|_{\mathrm{F}} = \rho_\ell$ deterministically. Conditionally on $\mathcal{F}_t$, the factors $(\boldsymbol{u}_1, \dots, \boldsymbol{u}_N, \boldsymbol{z}_0)$ are mutually independent. For fixed $\boldsymbol{g} \in T_\ell$ and $\boldsymbol{h} \in \mathbb{R}^{d_0}$:*

$$\mathbb{E}\big[\langle \boldsymbol{g}, \boldsymbol{u}_\ell \rangle \boldsymbol{u}_\ell\big] = \frac{\boldsymbol{g}}{d_\ell}, \qquad \textit{hence} \qquad \mathbb{E}\big[\langle \boldsymbol{g}, \widetilde{\boldsymbol{Z}}_\ell \rangle \widetilde{\boldsymbol{Z}}_\ell\big] = c_\ell \, \boldsymbol{g}, \qquad \mathbb{E}\big[\langle \boldsymbol{g}, \widetilde{\boldsymbol{Z}}_\ell \rangle^2\big] = c_\ell \|\boldsymbol{g}\|_{\mathrm{F}}^2, \tag{26}$$

$$\mathbb{E}\big[\langle \boldsymbol{h}, \boldsymbol{z}_0 \rangle \boldsymbol{z}_0\big] = \boldsymbol{h}, \qquad \mathbb{E}\big[\langle \boldsymbol{h}, \boldsymbol{z}_0 \rangle^2\big] = \|\boldsymbol{h}\|^2, \qquad \mathbb{E}\big[\langle \boldsymbol{h}, \boldsymbol{z}_0 \rangle^2 \|\boldsymbol{z}_0\|^2\big] = (d_0 + 2)\|\boldsymbol{h}\|^2, \tag{27}$$

$$\mathbb{E}\|\boldsymbol{z}_0\|^{2k} = d_0(d_0+2)\cdots(d_0+2k-2) \le (d_0 + 2k - 2)^k, \qquad k \ge 1. \tag{28}$$

With $Q := \sum_\ell \rho_\ell^2$ (deterministic), $\|\boldsymbol{u}_t\|^2 = Q + \|\boldsymbol{z}_0\|^2$, and consequently

$$\mathbb{E}\|\boldsymbol{u}_t\|^4 \le (\bar{D} + 2)^2, \qquad \mathbb{E}\|\boldsymbol{u}_t\|^8 \le (\bar{D} + 6)^4. \tag{29}$$

Proof. The orthogonal projection of an isotropic Gaussian onto a subspace is a standard Gaussian on that subspace, and normalizing a standard Gaussian on T_ℓ yields the uniform distribution on its unit sphere; scaling by ρ_ℓ gives the sphere of radius ρ_ℓ. Mutual independence holds because the blocks use independent Gaussians and the projectors are fixed under the conditioning. For the sphere moments, isotropy of $\boldsymbol{u}_\ell$ on T_ℓ implies that the covariance operator $\boldsymbol{g} \mapsto \mathbb{E}[\langle \boldsymbol{g}, \boldsymbol{u}_\ell \rangle \boldsymbol{u}_\ell]$ is a multiple of the identity on T_ℓ; taking the trace and using $\|\boldsymbol{u}_\ell\| = 1$ identifies the multiple as $1/d_\ell$, which gives the first identity in (26), and taking the inner product with $\boldsymbol{g}$ gives the second; multiplying by $\rho_\ell^2 = c_\ell d_\ell$ converts both to $\widetilde{\boldsymbol{Z}}_\ell$. The Gaussian identities in (27) are classical; for the third, expand $\sum_{i,k} h_i h_k \, \mathbb{E}[z_i z_k \|\boldsymbol{z}\|^2]$ and use $\mathbb{E} z_1^4 = 3$ and $\mathbb{E} z_1^2 z_j^2 = 1$ for $j \ne 1$. For (28), $\|\boldsymbol{z}_0\|^2 \sim \chi^2_{d_0}$ has $\mathbb{E}(\chi^2_{d_0})^k = d_0(d_0+2)\cdots(d_0+2k-2)$, and each factor is at most $d_0 + 2k - 2$. For (29), expand $\mathbb{E}(Q + X)^k$ with $X := \|\boldsymbol{z}_0\|^2$ binomially; each $\mathbb{E}X^j \le (d_0 + 2k - 2)^j$ for $j \le k$, so the binomial sum is at most $(Q + d_0 + 2k - 2)^k = (\bar{D} + 2k - 2)^k$; take $k = 2$ and $k = 4$. □

Lemma 2 (two-point bias). *Under Assumption 2, with $s_t^{\mathrm{lin}} := \langle \nabla F(\boldsymbol{x}_t; \xi_t), \boldsymbol{u}_t \rangle$,*

$$\big|\hat{s}_t - s_t^{\mathrm{lin}}\big| \le \frac{\rho_3}{6} \, \epsilon^2 \, \|\boldsymbol{u}_t\|^3. \tag{30}$$

Proof. By Assumption 2, $\phi(\tau) := F(\boldsymbol{x}_t + \tau \boldsymbol{u}_t; \xi_t)$ is three times differentiable on $[-\epsilon, \epsilon]$ (it is C^2 with ϕ''' existing pointwise, which suffices for the Lagrange remainder). Taylor expansion gives $\phi(\pm\epsilon) = \phi(0) \pm \epsilon\phi'(0) + \frac{\epsilon^2}{2}\phi''(0) \pm \frac{\epsilon^3}{6}\phi'''(\tau_\pm)$ for some $\tau_\pm \in (-\epsilon, \epsilon)$. The symmetric difference cancels $\phi''(0)$: $\hat{s}_t = \phi'(0) + \frac{\epsilon^2}{12}\big(\phi'''(\tau_+) + \phi'''(\tau_-)\big)$, and $|\phi'''| \le \rho_3 \|\boldsymbol{u}_t\|^3$. □

Remark 4 (structure of the bias). *The constant $1/6$ is sharp: it is attained by $\phi(\tau) = \tau^3$. Because the implementation evaluates the loss on the ambient tangent line rather than along a retracted curve, no retraction curvature enters this bound: it is the classical Euclidean symmetric-difference bound. First-order geometry (tangent projection, d_ℓ, c_ℓ) enters the analysis through the random-direction law of Lemma 1; second-order manifold geometry enters* only *through the retraction step (Lemmas 5 and 6). Note also that the single scalar $\hat{s}_t$ carries the finite-difference bias of all $N + 1$ factors at once — the coupling induced by the shared two-point evaluation — which is why the bias constant in Lemma 3 involves the total dimension $\bar{D}$ rather than a per-block quantity.*

Lemma 3 (first moment). *Under Assumptions 1–3,*

$$\mathbb{E}\big[\widehat{\boldsymbol{G}}_t \,\big|\, \mathcal{F}_t\big] = \boldsymbol{D} \, \mathrm{grad}\, f(\boldsymbol{x}_t) + \boldsymbol{\beta}_t, \qquad \|\boldsymbol{\beta}_t\| \le B_\epsilon := \frac{\rho_3}{6} \, \epsilon^2 \, (\bar{D} + 2)^2. \tag{31}$$

Proof. Condition on $\mathcal{F}_t$ and ξ_t. Write $\hat{s}_t = s_t^{\mathrm{lin}} + \delta_t$ with $|\delta_t| \le \frac{\rho_3}{6}\epsilon^2 \|\boldsymbol{u}_t\|^3$ by Lemma 2, and decompose $s_t^{\mathrm{lin}} = \sum_{k=1}^N a_k + a_0$ with $a_k := \langle P_k \nabla_k F, \widetilde{\boldsymbol{Z}}_k \rangle$ (tangency of $\widetilde{\boldsymbol{Z}}_k$ lets us insert P_k) and $a_0 := \langle \nabla_0 F, \boldsymbol{z}_0 \rangle$.

Component ℓ. $\mathbb{E}[s_t^{\mathrm{lin}} \widetilde{\boldsymbol{Z}}_\ell] = \mathbb{E}[a_\ell \widetilde{\boldsymbol{Z}}_\ell]$: for $k \ne \ell$, $\mathbb{E}[a_k \widetilde{\boldsymbol{Z}}_\ell] = \mathbb{E}[a_k] \, \mathbb{E}[\widetilde{\boldsymbol{Z}}_\ell]$ by conditional independence (Lemma 1), and both factors are zero-mean, so the cross terms vanish. Lemma 1 then

gives $\mathbb{E}[a_\ell \widetilde{\boldsymbol{Z}}_\ell] = c_\ell P_\ell \nabla_\ell F$. *Component* 0: $\mathbb{E}[s_t^{\mathrm{lin}} \boldsymbol{z}_0] = \nabla_0 F$ by the same argument. Hence $\mathbb{E}[s_t^{\mathrm{lin}} \boldsymbol{u}_t] = \boldsymbol{D} \operatorname{grad} F(\boldsymbol{x}_t; \xi_t)$.

Bias. $\|\mathbb{E}[\delta_t \boldsymbol{u}_t]\| \le \mathbb{E}\big[|\delta_t| \, \|\boldsymbol{u}_t\|\big] \le \frac{\rho_3}{6}\epsilon^2 \, \mathbb{E}\|\boldsymbol{u}_t\|^4 \le \frac{\rho_3}{6}\epsilon^2(\bar{D}+2)^2$ by Lemma 1. Finally take $\mathbb{E}_\xi$, using Assumption 3 and $P_\ell \, \mathbb{E}_\xi \nabla_\ell F = P_\ell \nabla_\ell f$. □

Remark 5 (interpretation). *The implemented estimator is an (ϵ-biased) stochastic* preconditioned *Riemannian gradient: the preconditioner $\boldsymbol{D}$ has per-factor gains $c_\ell = m_\ell n_\ell / d_\ell \ge 1$ on 2D blocks and exactly 1 on the 1D factor. The post-projection normalization to $\|\widetilde{\boldsymbol{Z}}_\ell\|_{\mathrm{F}} = \sqrt{m_\ell n_\ell}$ is what produces c_ℓ; the single-block specialization has $N = 1$ and $d_0 = 0$, for which $\bar{D} = mn$ and the bias bound reads $(\rho_3/6)\epsilon^2(mn)^2$ since $\|\boldsymbol{u}_t\| = \sqrt{mn}$ is deterministic — the $(\bar{D}+2)^2$ constant is not needed in that case. $\boldsymbol{D}$ is a fixed, deterministic, dimension-derived rescaling; it involves no tuning.*

Lemma 4 (second moment). *Under Assumptions 1–3,*

$$\mathbb{E}\big[\|\widehat{\boldsymbol{G}}_t\|^2 \,\big|\, \mathcal{F}_t\big] \le 2(\bar{D}+2)\big(\|\operatorname{grad} f(\boldsymbol{x}_t)\|_{\boldsymbol{D}}^2 + \sigma_D^2\big) + \frac{\rho_3^2}{18}\,\epsilon^4\,(\bar{D}+6)^4. \tag{32}$$

Proof. $\mathbb{E}[\|\widehat{\boldsymbol{G}}_t\|^2] = \mathbb{E}[\hat{s}_t^2 \|\boldsymbol{u}_t\|^2]$ and $\hat{s}_t^2 \le 2(s_t^{\mathrm{lin}})^2 + 2\delta_t^2$. Condition on $\mathcal{F}_t$ and ξ_t throughout; write $s_t^{\mathrm{lin}} = A + a_0$ with $A := \sum_\ell a_\ell$, which is conditionally independent of $\boldsymbol{z}_0$.

(i) With $\|\boldsymbol{u}_t\|^2 = Q + \|\boldsymbol{z}_0\|^2$,

$$\begin{aligned}
\mathbb{E}\big[(s_t^{\mathrm{lin}})^2 (Q + \|\boldsymbol{z}_0\|^2)\big] &= Q\,\mathbb{E}[(s_t^{\mathrm{lin}})^2] + \mathbb{E}[A^2]\,\mathbb{E}\|\boldsymbol{z}_0\|^2 + 2\,\mathbb{E}[A]\,\mathbb{E}[a_0\|\boldsymbol{z}_0\|^2] + \mathbb{E}[a_0^2\|\boldsymbol{z}_0\|^2] \\
&= Q\,S + d_0 \sum_\ell c_\ell \|P_\ell \nabla_\ell F\|_{\mathrm{F}}^2 + 0 + (d_0+2)\|\nabla_0 F\|^2 \\
&\le (Q + d_0 + 2)\,S = (\bar{D}+2)\,S,
\end{aligned}$$

where $S := \mathbb{E}[(s_t^{\mathrm{lin}})^2] = \sum_\ell c_\ell \|P_\ell \nabla_\ell F\|_{\mathrm{F}}^2 + \|\nabla_0 F\|^2 = \|\operatorname{grad} F(\boldsymbol{x}_t; \xi_t)\|_{\boldsymbol{D}}^2$ (Lemma 1; cross terms vanish as in Lemma 3, and $\mathbb{E}[A] = 0$ kills the middle term). The final inequality follows by merging the last two terms: since $\sum_\ell c_\ell \|P_\ell \nabla_\ell F\|_{\mathrm{F}}^2 = S - \|\nabla_0 F\|^2$, the sum equals $Q\,S + d_0\,S + 2\|\nabla_0 F\|^2 \le (Q + d_0 + 2)\,S$, using $\|\nabla_0 F\|^2 \le S$.

(ii) $\mathbb{E}[\delta_t^2 \|\boldsymbol{u}_t\|^2] \le \big(\frac{\rho_3}{6}\big)^2 \epsilon^4 \, \mathbb{E}\|\boldsymbol{u}_t\|^8 \le \frac{\rho_3^2}{36}\epsilon^4(\bar{D}+6)^4$ by Lemmas 2 and 1.

Combining, $\mathbb{E}[\|\widehat{\boldsymbol{G}}_t\|^2] \le 2(\bar{D}+2)\,\mathbb{E}_\xi\|\operatorname{grad} F\|_{\boldsymbol{D}}^2 + \frac{\rho_3^2}{18}\epsilon^4(\bar{D}+6)^4$. Finally, per factor, $\mathbb{E}_\xi\|P_\ell \nabla_\ell F\|^2 = \|P_\ell \nabla_\ell f\|^2 + \mathbb{E}_\xi\|P_\ell(\nabla_\ell F - \nabla_\ell f)\|^2$ (the cross term vanishes by Assumption 3 since $\mathbb{E}_\xi P_\ell(\nabla_\ell F - \nabla_\ell f) = 0$), so $\mathbb{E}_\xi\|\operatorname{grad} F\|_{\boldsymbol{D}}^2 \le \|\operatorname{grad} f\|_{\boldsymbol{D}}^2 + \sigma_D^2$. □

Lemma 5 (retraction error). *Let $\boldsymbol{X} \in \mathcal{M}_\ell$ with $\sigma_{r_\ell}(\boldsymbol{X}) \ge \sigma_{\min}$ and $\boldsymbol{Y} \in T_{\boldsymbol{X}}\mathcal{M}_\ell$ with $\|\boldsymbol{Y}\|_{\mathrm{F}} \le \sigma_{\min}/2$. Assume $m_\ell, n_\ell \ge 2r_\ell$ (our settings have $m_\ell, n_\ell \ge 2048 \gg 2r_\ell$; the degenerate case is discussed in the proof). Then*

$$\big\|\mathcal{H}_{r_\ell}(\boldsymbol{X}+\boldsymbol{Y}) - (\boldsymbol{X}+\boldsymbol{Y})\big\|_{\mathrm{F}} \le \frac{\|\boldsymbol{Y}\|_{\mathrm{F}}^2}{\sigma_{\min}}. \tag{33}$$

On the Euclidean factor the retraction is the identity, with zero error.

Proof. Drop the block index. Write $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ (thin SVD) and decompose the tangent vector as $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{M}\boldsymbol{V}^\top + \boldsymbol{U}_p\boldsymbol{V}^\top + \boldsymbol{U}\boldsymbol{V}_p^\top$ with $\boldsymbol{U}^\top\boldsymbol{U}_p = 0$ and $\boldsymbol{V}^\top\boldsymbol{V}_p = 0$. The three components are mutually Frobenius-orthogonal (all cross traces vanish via $\boldsymbol{U}^\top\boldsymbol{U}_p = 0$, $\boldsymbol{V}^\top\boldsymbol{V}_p = 0$), so

$$\|\boldsymbol{Y}\|_{\mathrm{F}}^2 = \|\boldsymbol{M}\|_{\mathrm{F}}^2 + \|\boldsymbol{U}_p\boldsymbol{V}^\top\|_{\mathrm{F}}^2 + \|\boldsymbol{U}\boldsymbol{V}_p^\top\|_{\mathrm{F}}^2. \tag{34}$$

Choose $\boldsymbol{Q}_U$ as any r orthonormal columns of $\operatorname{col}(\boldsymbol{U})^\perp$ whose span contains $\operatorname{col}(\boldsymbol{U}_p)$ (if $m < 2r$, the second block shrinks to $\dim \le m - r$ columns and the argument below is unchanged); choose $\boldsymbol{Q}_V$ analogously. In the extended bases $[\boldsymbol{U}, \boldsymbol{Q}_U]$ and $[\boldsymbol{V}, \boldsymbol{Q}_V]$,

$$\boldsymbol{X} + \boldsymbol{Y} = [\boldsymbol{U}, \boldsymbol{Q}_U]\,\boldsymbol{K}\,[\boldsymbol{V}, \boldsymbol{Q}_V]^\top, \qquad \boldsymbol{K} = \begin{bmatrix} \boldsymbol{\Sigma} + \boldsymbol{M} & \boldsymbol{R}_1 \\ \boldsymbol{R}_2 & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{2r \times 2r},$$

with $\|\boldsymbol{R}_1\|_{\mathrm{F}} = \|\boldsymbol{U}\boldsymbol{V}_p^\top\|_{\mathrm{F}}$ and $\|\boldsymbol{R}_2\|_{\mathrm{F}} = \|\boldsymbol{U}_p\boldsymbol{V}^\top\|_{\mathrm{F}}$ (the orthonormal-basis factors preserve the Frobenius norms of the two normal components). Whenever $\boldsymbol{U}_p$ and $\boldsymbol{V}_p$ have full column rank r — an almost-sure event for the Gaussian-derived tangent random matrices — this $\boldsymbol{K}$ is exactly the $2r \times 2r$ core assembled by the efficient retraction of Section 4.3.

Since $\|\boldsymbol{M}\|_2 \le \|\boldsymbol{M}\|_{\mathrm{F}} \le \|\boldsymbol{Y}\|_{\mathrm{F}} \le \sigma_{\min}/2$, Weyl's inequality gives $\sigma_{\min}(\boldsymbol{\Sigma}+\boldsymbol{M}) \ge \sigma_{\min}/2$, i.e. $\|(\boldsymbol{\Sigma}+\boldsymbol{M})^{-1}\|_2 \le 2/\sigma_{\min}$. Define the rank-$\le r$ competitor

$$\boldsymbol{K}_0 := \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{R}_2(\boldsymbol{\Sigma}+\boldsymbol{M})^{-1} \end{bmatrix} (\boldsymbol{\Sigma}+\boldsymbol{M}) \begin{bmatrix} \boldsymbol{I} & (\boldsymbol{\Sigma}+\boldsymbol{M})^{-1}\boldsymbol{R}_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}+\boldsymbol{M} & \boldsymbol{R}_1 \\ \boldsymbol{R}_2 & \boldsymbol{R}_2(\boldsymbol{\Sigma}+\boldsymbol{M})^{-1}\boldsymbol{R}_1 \end{bmatrix}.$$

Because $\mathcal{H}_r(\boldsymbol{K})$ is the best rank-r approximation of $\boldsymbol{K}$ in Frobenius norm and $\boldsymbol{K}_0$ has rank at most r,

$$\begin{aligned} \|\boldsymbol{K} - \mathcal{H}_r(\boldsymbol{K})\|_{\mathrm{F}} \le \|\boldsymbol{K} - \boldsymbol{K}_0\|_{\mathrm{F}} &= \left\|\boldsymbol{R}_2(\boldsymbol{\Sigma}+\boldsymbol{M})^{-1}\boldsymbol{R}_1\right\|_{\mathrm{F}} \\ &\le \|\boldsymbol{R}_2\|_{\mathrm{F}} \left\|(\boldsymbol{\Sigma}+\boldsymbol{M})^{-1}\right\|_2 \|\boldsymbol{R}_1\|_{\mathrm{F}} \le \frac{2}{\sigma_{\min}} \|\boldsymbol{R}_1\|_{\mathrm{F}}\|\boldsymbol{R}_2\|_{\mathrm{F}}. \end{aligned}$$

By the AM–GM inequality and (34), $\|\boldsymbol{R}_1\|_{\mathrm{F}}\|\boldsymbol{R}_2\|_{\mathrm{F}} \le \frac{1}{2}(\|\boldsymbol{R}_1\|_{\mathrm{F}}^2 + \|\boldsymbol{R}_2\|_{\mathrm{F}}^2) \le \frac{1}{2}\|\boldsymbol{Y}\|_{\mathrm{F}}^2$, so $\|\boldsymbol{K} - \mathcal{H}_r(\boldsymbol{K})\|_{\mathrm{F}} \le \|\boldsymbol{Y}\|_{\mathrm{F}}^2/\sigma_{\min}$. Since $[\boldsymbol{U}, \boldsymbol{Q}_U]$ and $[\boldsymbol{V}, \boldsymbol{Q}_V]$ have orthonormal columns, $\boldsymbol{X}+\boldsymbol{Y}$ and $\boldsymbol{K}$ share the same nonzero singular values, so their best rank-r approximation errors coincide, which transfers the bound from $\boldsymbol{K}$ to $\boldsymbol{X}+\boldsymbol{Y}$. □

Under Assumption 1(iv), the radius condition $\|\eta\widehat{\boldsymbol{G}}_{t,\ell}\|_{\mathrm{F}} \le \sigma_{\min}/2$ holds at every step and 2D block, so Lemma 5 applies to every realized update.

Lemma 6 (descent inequality)**.** *Under Assumptions 1–3, with $L_R := L + 2G_\infty/\sigma_{\min}$, for every step t, almost surely,*

$$f(\boldsymbol{x}_{t+1}) \le f(\boldsymbol{x}_t) - \eta\left\langle \operatorname{grad} f(\boldsymbol{x}_t), \widehat{\boldsymbol{G}}_t\right\rangle + \frac{L_R}{2}\eta^2\|\widehat{\boldsymbol{G}}_t\|^2. \tag{35}$$

Proof. (i) L-smoothness of f (the consequence of Assumptions 2 and 3 noted in Appendix C.2) along the ambient update segment $[\boldsymbol{x}_t, \boldsymbol{x}_t - \eta\widehat{\boldsymbol{G}}_t] \subset \mathcal{X}'_{\mathrm{reg}}$ gives $f(\boldsymbol{x}_t - \eta\widehat{\boldsymbol{G}}_t) \le f(\boldsymbol{x}_t) - \eta\langle\nabla f(\boldsymbol{x}_t), \widehat{\boldsymbol{G}}_t\rangle + \frac{L}{2}\eta^2\|\widehat{\boldsymbol{G}}_t\|^2$; every component of $\widehat{\boldsymbol{G}}_t$ is tangent, so $\langle\nabla f, \widehat{\boldsymbol{G}}_t\rangle = \langle\operatorname{grad} f, \widehat{\boldsymbol{G}}_t\rangle$. (ii) Retraction correction:

$$\begin{aligned} \left\|\boldsymbol{x}_{t+1} - (\boldsymbol{x}_t - \eta\widehat{\boldsymbol{G}}_t)\right\| &= \Big(\sum_\ell \left\|\mathcal{H}_{r_\ell}(\Delta\boldsymbol{W}_{\ell,t} - \eta\widehat{\boldsymbol{G}}_{t,\ell}) - (\Delta\boldsymbol{W}_{\ell,t} - \eta\widehat{\boldsymbol{G}}_{t,\ell})\right\|_{\mathrm{F}}^2\Big)^{1/2} \\ &\le \frac{\eta^2}{\sigma_{\min}}\Big(\sum_\ell \|\widehat{\boldsymbol{G}}_{t,\ell}\|_{\mathrm{F}}^4\Big)^{1/2} \le \frac{\eta^2}{\sigma_{\min}}\|\widehat{\boldsymbol{G}}_t\|^2, \end{aligned}$$

using Lemma 5 with Assumption 1(iv), and $\sqrt{\sum_\ell a_\ell^2} \le \sum_\ell a_\ell$ for $a_\ell \ge 0$. The mean value theorem on the third branch of $\mathcal{X}'_{\mathrm{reg}}$ with Assumption 1(ii) gives $f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t - \eta\widehat{\boldsymbol{G}}_t) \le G_\infty\,(\eta^2/\sigma_{\min})\|\widehat{\boldsymbol{G}}_t\|^2$. Summing (i) and (ii) yields the coefficient $L/2 + G_\infty/\sigma_{\min} = L_R/2$. □

C.3 Formal convergence theorem and proof

The following is the formal version of Theorem 1.

Theorem 2 (Conditional convergence of ZO-RGD, formal)**.** *Let Assumptions 1–3 hold, and let $L_R := L + 2G_\infty/\sigma_{\min}$, with $\bar{D}$, σ_D^2, and Δ_0 as defined in Appendices C.1 and C.2. Run base ZO-RGD (Algorithm 1) for T steps with the constant stepsize*

$$\eta = \min\left\{\frac{1}{2L_R(\bar{D}+2)},\ \sqrt{\frac{\Delta_0}{L_R(\bar{D}+2)\sigma_D^2 T}}\right\}, \tag{36}$$

where the second candidate is read as $+\infty$ if $\sigma_D = 0$. Then

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[\|\operatorname{grad} f(\boldsymbol{x}_t)\|_{\boldsymbol{D}}^2\big] \le 8\sqrt{\frac{\Delta_0 L_R(\bar{D}+2)\sigma_D^2}{T}} + \frac{8\Delta_0 L_R(\bar{D}+2)}{T} + \frac{\rho_3^2\epsilon^4}{9}\left[(\bar{D}+2)^4 + \frac{(\bar{D}+6)^4}{2(\bar{D}+2)}\right]. \tag{37}$$

Since $\|\boldsymbol{v}\| \le \|\boldsymbol{v}\|_{\boldsymbol{D}}$ for every tangent vector $\boldsymbol{v}$, the same right-hand side bounds the average of $\mathbb{E}\|\operatorname{grad} f(\boldsymbol{x}_t)\|^2$.

Proof. Take $\mathbb{E}[\,\cdot \mid \mathcal{F}_t]$ of Lemma 6 and insert Lemmas 3 and 4:

$$\mathbb{E}[f(\boldsymbol{x}_{t+1}) \mid \mathcal{F}_t] \le f(\boldsymbol{x}_t) - \eta\big\langle \operatorname{grad} f, \boldsymbol{D}\operatorname{grad} f + \boldsymbol{\beta}_t\big\rangle + \frac{L_R}{2}\eta^2\Big[2(\bar{D}+2)\big(\|\operatorname{grad} f\|_{\boldsymbol{D}}^2 + \sigma_D^2\big) + \frac{\rho_3^2}{18}\epsilon^4(\bar{D}+6)^4\Big],$$

where all gradients are evaluated at $\boldsymbol{x}_t$. The leading term is $-\eta\langle \operatorname{grad} f, \boldsymbol{D}\operatorname{grad} f\rangle = -\eta\|\operatorname{grad} f\|_{\boldsymbol{D}}^2$. For the cross term, Young's inequality ($ab \le a^2/4 + b^2$) with $\|\operatorname{grad} f\| \le \|\operatorname{grad} f\|_{\boldsymbol{D}}$ gives $-\eta\langle \operatorname{grad} f, \boldsymbol{\beta}_t\rangle \le \eta\|\operatorname{grad} f\| B_\epsilon \le \frac{\eta}{4}\|\operatorname{grad} f\|_{\boldsymbol{D}}^2 + \eta B_\epsilon^2$. The condition $\eta \le 1/(2L_R(\bar{D}+2))$ makes $L_R\eta^2(\bar{D}+2)\|\operatorname{grad} f\|_{\boldsymbol{D}}^2 \le \frac{\eta}{2}\|\operatorname{grad} f\|_{\boldsymbol{D}}^2$. Hence

$$\mathbb{E}[f(\boldsymbol{x}_{t+1}) \mid \mathcal{F}_t] \le f(\boldsymbol{x}_t) - \frac{\eta}{4}\|\operatorname{grad} f(\boldsymbol{x}_t)\|_{\boldsymbol{D}}^2 + \eta B_\epsilon^2 + \eta^2 L_R(\bar{D}+2)\sigma_D^2 + \eta^2\frac{L_R\rho_3^2}{36}\epsilon^4(\bar{D}+6)^4.$$

Telescoping over $t = 0, \dots, T-1$, taking full expectations, using $f \ge f_*$, and multiplying by $4/(\eta T)$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\operatorname{grad} f(\boldsymbol{x}_t)\|_{\boldsymbol{D}}^2 \le \frac{4\Delta_0}{\eta T} + 4B_\epsilon^2 + 4\eta L_R(\bar{D}+2)\sigma_D^2 + \eta\,\frac{L_R\rho_3^2}{9}\,\epsilon^4(\bar{D}+6)^4. \tag{38}$$

With $\eta = \min\big\{1/(2L_R(\bar{D}+2)), \sqrt{\Delta_0/(L_R(\bar{D}+2)\sigma_D^2 T)}\big\}$, the standard argument ($1/\min \le$ sum of reciprocals, and the second candidate balances the first and third terms of (38)) gives

$$\frac{4\Delta_0}{\eta T} + 4\eta L_R(\bar{D}+2)\sigma_D^2 \le 8\sqrt{\frac{\Delta_0 L_R(\bar{D}+2)\sigma_D^2}{T}} + \frac{8\Delta_0 L_R(\bar{D}+2)}{T},$$

which gives the first two terms of (37). For the remaining two terms of (38), the definition of B_ϵ gives $4B_\epsilon^2 = \frac{\rho_3^2}{9}\epsilon^4(\bar{D}+2)^4$, and $\eta \le 1/(2L_R(\bar{D}+2))$ gives $\eta\,\frac{L_R\rho_3^2}{9}\epsilon^4(\bar{D}+6)^4 \le \frac{\rho_3^2}{18}\epsilon^4(\bar{D}+6)^4/(\bar{D}+2)$. Summing yields (37). Since $\|\boldsymbol{v}\| \le \|\boldsymbol{v}\|_{\boldsymbol{D}}$ for every tangent vector, the same right-hand side bounds the plain-norm average. □

Remark 6 (closing the step-regularity condition deterministically)**.** *Theorem 2 assumes that the realized steps of the chosen η satisfy Assumption 1(iv). This can be closed deterministically at the price of an extra $O(1/T)$ term: under the sufficient condition of Remark 3(b), set $\eta_{\mathrm{reg}} := \epsilon\sigma_{\min}/(2B_F\max_\ell \rho_\ell)$ and choose the three-way minimum $\eta = \min\{1/(2L_R(\bar{D}+2)), \sqrt{\Delta_0/(L_R(\bar{D}+2)\sigma_D^2 T)}, \eta_{\mathrm{reg}}\}$. Since $1/\min \le$ the sum of reciprocals, the $4\Delta_0/(\eta T)$ term contributes at most one extra summand $4\Delta_0/(\eta_{\mathrm{reg}}T) = 8\Delta_0 B_F\max_\ell\rho_\ell/(\epsilon\sigma_{\min}T)$, which is $O(1/T)$ and does not affect the leading $\sqrt{1/T}$ rate.*

Remark 7 (function-evaluation complexity and the ϵ-condition)**.** *Each step uses two forward passes. In the regime $\epsilon_s^2 \le \sigma_D^2$, to reach $(1/T)\sum_t \mathbb{E}\|\operatorname{grad} f(\boldsymbol{x}_t)\|^2 \le \epsilon_s^2$ it suffices to take $T = O\big(\Delta_0 L_R(\bar{D}+2)\sigma_D^2/\epsilon_s^4\big)$ together with $\epsilon^2 \le 3\epsilon_s/(\sqrt{2}\,\rho_3(\bar{D}+2)^2)$, which makes $4B_\epsilon^2 \le \epsilon_s^2/2$ and renders the ϵ^4 term subdominant. The ϵ-condition carries ambient-dimension powers, as in standard ambient ZO analyses (e.g., the μ-condition in the MeZO analysis (Malladi et al., 2023)); ρ_3 is a worst-case directional constant along the realized segments. We state the sufficient scaling explicitly rather than absorbing it into asymptotic notation.*

Table 7: Unified-protocol configuration card. All methods use two forward passes per step; ranks are each method's published setting; the shared grid is the MeZO grid, verbatim; selection is development-loss only (grid/pilot seed disjoint from report seeds).

Panel	Arms	Rank	LR selection	Steps × batch	Report seeds
OPT-13B controlled	MeZO, LOZO, ZO-RGD, ZO-RGD-S, LoRA+AdamW	$-, 1, 4, 2, 8$	shared grid, dev loss, seed 0	$20{,}000 \times 16$	$\{1, 2, 3\}$
OPT-1.3B breadth / mechanism	6 forward-only methods / controls	published	shared grid, dev loss, seed 0	$20{,}000 \times 16$	$\{1, 2, 3\}$ (breadth: single fresh seed)
Modern panel	ZO-RGD, ZO-RGD-S, LOZO	$4, 2, 1$	per-method dev-only pilot, seed 7	$20{,}000 \times 16$	$\{0, 1, 2\}$

Table 8: Baseline implementation audit for the unified-protocol tables. All baselines run through the same data, prompt, and evaluation path as our methods, from the local baseline implementations, with the deviations column exhaustive.

Method	Estimator setting	Tuned over	Deviations from official
MeZO	full-parameter ZO, $q = 1$	shared MeZO grid	none
LOZO	official $r = 1$, $\nu = 100$	shared MeZO grid	none (an earlier internal matrix used ν=50)
MeZO-LoRA / LOZO+LoRA	LoRA $r = 8, \alpha = 16$	official LoRA grid (LR $\{10^{-4}, 5{\times}10^{-5}\}$, $+10^{-5}$ for QA; $\epsilon = 10^{-2}$)	none
LoRA+AdamW (first-order ref.)	LoRA $r = 8, \alpha = 16$, AdamW	LR $\{10^{-5}, 5{\times}10^{-5}, 10^{-4}\}$, dev loss	not a forward-only method; reference row only

D Additional Experimental Details

This appendix follows the order of Section 6. Appendices D.1 and D.2 give the training protocol and compute resources. Appendices D.3–D.6 supplement the broad-suite comparison with the complete OPT-1.3B table, its unified-protocol rerun, the trainability floor, and the OPT-30B spot-check. Appendices D.7–D.9 supplement the controlled comparison with per-seed values, the matched-rank comparison, and a shared-learning-rate regime with convergence against forward passes. Appendix D.10 gives the learning-rate diagnostic for the modern-model panel. Appendices D.11–D.15 collect the geometry controls, negative results, ablations, and stability diagnostics; Appendices D.16 and D.17 give runtime and memory; and Appendix D.18 is the failure register.

D.1 Training protocol and configurations

For ZO-RGD and ZO-RGD-S, each ZO update uses one normalized tangent random matrix, so the two-point finite-difference scalar is obtained with two forward passes. Unless otherwise noted, runs use FP16 checkpoints, $20{,}000$ training steps, constant learning rate, $\epsilon = 10^{-3}$, and weight decay 0. Seeds are paired: the training-subset seed and the trainer seed move together. Hyperparameter selection uses a dedicated grid seed (0) on OPT and a disjoint pilot seed (7) on the modern-model panel; report seeds are $\{1, 2, 3\}$ on the OPT panels and $\{0, 1, 2\}$ on the modern-model panel (the two panels were executed independently; neither report set overlaps its selection seed, and the per-panel seed sets are disclosed here because they differ).

Each run samples three disjoint subsets with the paired seeds: up to 1000 training examples, up to 500 development examples held out from the training data (100 for CB and COPA), and up to 1000 evaluation examples from the task's validation split; smaller datasets are capped by the available split sizes, and reported n is the actual subset size of the run. Training operates only on the training subset. Within each run, the kept checkpoint is the one with the lowest development loss (`eval_loss`), under the same cadence for every method; development loss is the selection quantity because it is the only development-set signal produced by every trainer in the comparison, so a single selection rule can be applied symmetrically. The evaluation subset is read exactly once per run, after all selection. Table 7 summarizes the per-method configurations, and Table 8 audits the baseline implementations and their deviations.

Checkpoint conventions of the retained broad-suite tables. The three broad-suite tables of Section 6.2 retain their original reporting conventions, which differ from the unified protocol above. At OPT-1.3B (Table 9), our rows use best-development checkpoints while the four baseline rows report the final step. At OPT-13B (Table 1), our cells are selected by development metric within each variant's completed $20{,}000$-step sweep, while the baseline rows are cited from LOZO. At OPT-30B (Table 2), the cited rows follow the LOZO reporting and our cells were not selected under the unified rule, so the table serves as scaling evidence only; the symmetric OPT-30B evidence is the BoolQ spot-check of Appendix D.6. Every controlled claim in the paper rests on the unified-protocol tables.

Table 9: OPT-1.3B complete local comparison on the eleven-task suite. Best and second-best results in each column are shown in **bold** and underlined, respectively.

Method	SST-2	RTE	CB	BoolQ	WSC	WiC	MultiRC	COPA	ReCoRD	SQuAD	DROP
MeZO	87.61	53.43	69.64	62.40	64.42	57.21	56.70	73.00	70.60	73.25	22.82
MeZO-LoRA	**90.94**	53.07	67.86	62.80	61.54	58.31	55.50	74.00	**72.20**	76.59	22.48
LOZO	88.88	58.12	66.07	62.20	**65.38**	56.43	56.10	74.00	70.20	77.00	22.23
LOZO+LoRA	89.45	54.87	69.64	64.00	61.54	**58.46**	56.10	74.00	71.70	74.76	22.39
ZO-RGD	90.83	59.57	71.43	**64.70**	61.54	58.31	57.20	75.00	71.80	76.61	24.51
ZO-RGD-S	90.48	**60.65**	**73.21**	64.60	61.54	58.15	**58.00**	**78.00**	71.50	**77.65**	**24.64**

Table 10: OPT-1.3B breadth, unified protocol, single fresh seed per cell (%). No bolding: single-seed cells are not ranked. SQuAD/DROP report F1.

Method	SST-2	RTE	CB	BoolQ	WSC	WiC	MultiRC	COPA	ReCoRD	SQuAD	DROP
MeZO	89.33	65.34	69.64	64.40	60.58	58.31	60.90	76.00	70.30	75.04	24.85
LOZO	87.61	65.34	67.86	64.70	61.54	60.03	57.20	73.00	70.20	72.71	23.27
MeZO-LoRA	90.02	63.54	69.64	65.00	61.54	58.62	60.30	74.00	70.50	75.84	24.82
LOZO+LoRA	90.71	63.54	71.43	65.20	58.65	58.93	60.10	74.00	71.90	75.84	25.55
ZO-RGD	92.43	65.34	71.43	66.90	55.77	57.84	59.40	79.00	68.60	77.66	26.30
ZO-RGD-S	92.55	63.90	71.43	65.30	59.62	61.60	59.60	70.00	67.70	77.96	25.97

D.2 Compute resources and reproducibility

ZO-RGD and ZO-RGD-S experiments were conducted on a shared academic GPU cluster with heterogeneous NVIDIA GPUs (V100, A100-80G, H100, H200); the unified-protocol OPT-13B cells ran on A100-80G and H200, with per-cell hardware recorded in the manifest. Wall-clock is compared only within one GPU model (Appendix D.16), and no cross-node timing ratio is used anywhere in the paper. Every reported cell carries a manifest row, job ID, result file, configuration, selection rule, seeds, hardware, status (including failures), and, for runs after provenance instrumentation landed, the exact code commit, released with the code together with the per-seed tables of Appendix D.7.

D.3 OPT-1.3B: complete forward-only comparison

Table 9 reports OPT-1.3B results on the full eleven-task suite. This scale provides a complete local comparison among forward-only methods, complementing the cited large-model tables of Section 6.2. It tests whether the fixed-rank additive-update formulation is already effective at a scale where the full baseline set can be evaluated. The table is retained under its original convention: our rows use best-development checkpoints and the baseline rows use the final step (Appendix D.1); the symmetric unified-protocol rerun of the same suite is Table 10 in Appendix D.4.

The two Riemannian ZO variants give the best forward-only result on 7 of the 11 tasks. The gains are distributed across task families, including entailment and semantic classification tasks such as RTE and CB, reasoning tasks such as MultiRC and COPA, and extractive QA tasks such as SQuAD and DROP. This result suggests that optimizing the persistent fixed-rank update is competitive across the complete suite, rather than benefiting only one benchmark type.

D.4 OPT-1.3B breadth under the unified protocol

Table 10 reruns the full eleven-task suite under the unified protocol (shared grid, development-loss selection, evaluation set read once), at a single fresh report seed per cell; it revisits the same eleven tasks as the retained Table 9 under the symmetric convention. Cells are single-seed and therefore not ranked statistically. A selection-stability check over the 66 (method, task) cells, formal fresh-seed value minus grid-seed value, gives mean -0.60, sd 2.31, with $|\Delta| > 4$ points only on small-evaluation-set tasks and in both directions, so the grid stage does not systematically inflate the fresh-seed numbers. Read alongside the trainability floor (Appendix D.5): on WiC, COPA, and ReCoRD the best trained method sits within two standard deviations of the untrained model, so differences there are mostly noise.

Table 11: Trainability floor at OPT-1.3B (%). Δ/sd normalizes the best-case trained gain by evaluation sampling noise; seed-to-seed training variance is not included, so every ratio is optimistic.

Task	n_{eval}	Zero-shot	Best trained	Δ best	Δ/sd
SQuAD	1000	27.67	77.65	+49.98	31.6
SST-2	872	53.56	90.94	+37.38	22.1
BoolQ	1000	47.03	64.70	+17.67	11.2
DROP	1000	11.23	24.64	+13.41	8.5
MultiRC	1000	46.00	58.00	+12.00	7.6
CB	56	37.50	73.21	+35.71	5.3
WSC	104	43.27	65.38	+22.11	4.5
RTE	277	53.07	60.65	+7.58	2.5
ReCoRD	1000	69.90	72.20	+2.30	1.5
COPA	100	75.00	78.00	+3.00	0.6
WiC	638	57.68	58.46	+0.78	0.4

D.5 Zero-shot trainability floor at OPT-1.3B

Table 11 tests whether each task responds to forward-only training at this scale. It compares the self-run untrained (zero-shot) score with the best trained cell of the historical eleven-task pool (over which the floor analysis was originally built), normalizing the best-case gain by evaluation sampling noise alone, so the ratio is an optimistic upper bound on signal. Three tasks fall below two standard deviations of the untrained model, ReCoRD (1.5 sd), COPA (0.6), WiC (0.4), and on COPA and WiC at least one trained row scores at or below the untrained model. Recomputing the ratios against the unified-protocol breadth table changes the numbers slightly (WiC 2.0, COPA 0.8, ReCoRD 1.3) and no conclusion: columns on these tasks measure noise more than method, and we do not interpret differences on them anywhere in this paper.

D.6 OPT-30B single-seed spot-check

The OPT-30B BoolQ column of Table 3 reports the unified-protocol BoolQ spot-check (shared grid, development-loss winner at 10^{-7} for all four methods, grid seed 0). An earlier pre-locked fixed-configuration triple is consistent (ZO-RGD/ZO-RGD-S/LOZO: 72.00/72.50/68.70). These cells are single-seed, carry no uncertainty estimate, and are not ranked statistically; they are the only symmetric 30B evidence and update the corresponding cells of the retained Table 2. The 30B SQuAD column was not run: all four methods exceed 80 GB at the first forward pass on A100-80G, and the available longer-wall-clock partitions did not admit the run.

D.7 Per-seed values and provenance

The controlled table compares matched self-run rows at OPT-13B and a uniform-convention OPT-30B spot-check. HiZOO is shown only for the verified three-seed RTE cell; unavailable matched cells are not inferred. Tables 12 and 13 report the per-seed values behind the controlled comparison and the modern-model panel. Untrained zero-shot reference evaluations under the same harness give RTE 60.29/59.57 (two runs) and SQuAD 46.23/48.35/46.59/48.19/46.29 (five runs). Every reported cell of this paper maps to one cluster job ID, one result file, and one manifest row (selection rule, learning rate, seeds, hardware, and status, including failed runs); the manifest is released with the code, and the per-seed values for the mechanism blocks of Table 5 are recorded there in the same format. The HiZOO dagger row uses RTE seeds $\{0, 1, 2\}$, with per-seed accuracies 61.37/59.93/64.98; a matched SQuAD run is unavailable.

D.8 Matched-rank comparison at OPT-13B

Table 14 reports the matched-rank comparison summarized in Section 6.3, and Table 15 gives the paired values behind it. Both variants use rank 2, seeds $\{1, 2, 3\}$, 20,000 updates, and development-loss checkpoint selection. The learning rates (RTE: 10^{-6}; SQuAD: 10^{-7}) were selected for ZO-RGD-S and fixed before the additional ZO-RGD runs; they were not independently tuned for ZO-RGD in this comparison.

Table 12: Per-seed values (%) for the non-dagger rows of the OPT-13B controlled comparison (Table 3), report seeds $\{1, 2, 3\}$. NR: no formal LoRA+AdamW SQuAD seed results; the A100 grid attempt ran out of memory.

	RTE (acc)			SQuAD (F1)		
Method	s1	s2	s3	s1	s2	s3
MeZO	58.84	61.73	60.65	82.67	81.88	83.77
LOZO	62.45	61.37	62.45	82.98	81.61	82.00
ZO-RGD	62.09	64.62	58.84	84.23	82.21	83.46
ZO-RGD-S	68.95	72.92	66.43	86.08	83.50	85.08
LoRA+AdamW	83.39	80.51	86.28	NR	NR	NR

Table 13: Per-seed values (%) for the modern-model panel (Table 4), report seeds $\{0, 1, 2\}$.

		ZO-RGD			ZO-RGD-S			LOZO		
Model	Task	s0	s1	s2	s0	s1	s2	s0	s1	s2
Llama-3.1-8B	RTE	61.37	66.06	59.93	70.40	53.79	64.62	58.84	49.10	62.82
Llama-3.1-8B	BoolQ	65.70	66.50	68.60	85.50	84.90	87.10	63.00	63.30	58.50
Llama-3.1-8B	SQuAD	87.34	86.19	87.62	84.42	86.44	85.52	88.77	84.39	87.13
Llama-3.1-8B	GSM8K	32.50	44.00	36.50	33.00	35.00	34.00	33.00	28.50	32.00
Qwen2.5-7B	RTE	87.36	88.81	88.81	87.73	89.53	87.73	84.48	88.09	87.36
Qwen2.5-7B	BoolQ	81.40	82.00	83.80	82.50	82.70	81.70	80.30	80.50	81.40
Qwen2.5-7B	SQuAD	88.16	87.73	87.66	87.72	87.44	87.70	86.23	86.84	87.80
Qwen2.5-7B	GSM8K	70.50	73.00	76.00	68.00	73.50	69.50	71.00	75.00	76.50

Table 14: Scalar smoothing at fixed rank and learning rate on OPT-13B. Both methods use $r = 2$ and paired seeds $\{1, 2, 3\}$. The rates were selected on the ZO-RGD-S development grid and fixed before the ZO-RGD runs. Scores are mean $\pm$ sd (%); all runs completed.

Task	Method	Learning rate	Score (mean $\pm$ sd)
RTE (acc.)	ZO-RGD	10^{-6}	64.50 ± 3.26
	ZO-RGD-S	10^{-6}	69.43 ± 3.28
SQuAD (F1)	ZO-RGD	10^{-7}	85.38 ± 0.34
	ZO-RGD-S	10^{-7}	84.89 ± 1.30

Table 15: Matched-rank OPT-13B scores (%). All twelve runs completed. Differences are ZO-RGD-S minus ZO-RGD, calculated before rounding.

Task	Seed	ZO-RGD	ZO-RGD-S	Difference
RTE	1	67.87	68.95	+1.08
	2	61.37	72.92	+11.55
	3	64.26	66.43	+2.17
SQuAD	1	85.67	86.08	+0.42
	2	85.00	83.50	-1.50
	3	85.48	85.08	-0.40

The mean paired differences are $+4.93$ points on RTE and -0.50 on SQuAD, with paired-difference standard deviations 5.76 and 0.96, respectively (two-sided paired t-tests: $p = 0.276$ and 0.466). All three RTE seeds favor ZO-RGD-S, but excluding seed 2 reduces the mean difference to $+1.62$. SQuAD favors ZO-RGD-S on one of three seeds. The observations are task-dependent; neither test establishes a difference, and nonsignificance does not establish equivalence.

D.9 Shared-learning-rate controlled core at OPT-13B

A companion regime to the shared-grid protocol of Section 6.1: a single pre-locked learning rate (10^{-7}, fixed in advance for every method and both tasks) applied identically to ZO-RGD, ZO-RGD-S, and LOZO, with no method receiving any tuning, on five paired seeds $\{0, \ldots, 4\}$. No method receives any tuning in this regime; its value is the paired structure. Two deviations from the main-

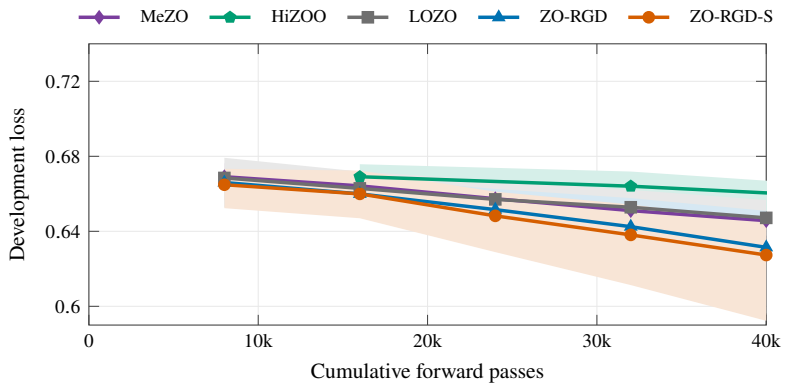


Figure 2: OPT-13B RTE development loss against cumulative forward passes at the shared learning rate 10^{-7}, truncated at the common 40K-forward budget. Lines are means and bands are ranges over completed runs. HiZOO uses four forward passes per step; the other methods use two.

Table 16: Shared-learning-rate core (%; mean $\pm$ sd over completed seeds; completed/attempted in parentheses). Paired differences vs. LOZO use only seeds where both methods completed; the sign test is exact and one-sided. "Strict separation" means the worst seed of ours beats the best seed of LOZO on the paired subset.

Method	RTE (acc)	SQuAD (F1)
LOZO	61.7 ± 1.8 (5/5)	81.8 ± 0.9 (5/5)
ZO-RGD	64.3 ± 5.8 (4/5)[a]	85.2 ± 1.0 (3/5)[b]
ZO-RGD-S	64.0 ± 4.5 (5/5)	85.0 ± 0.9 (5/5)

Paired difference vs. LOZO: ZO-RGD-S SQuAD $+3.2$, 5/5 wins, $p = 0.031$, strict separation; ZO-RGD SQuAD $+3.3$, 3/3 wins, $p = 0.125$, strict separation; RTE: both variants $p > 0.18$, no separation.

[a] One RTE seed was wall-clock truncated and excluded by the completion rule; [b] seeds 0 and 3 diverged (first non-finite diagnostic at steps 500 and 1500; Appendix D.18). ZO-RGD-S completed all five seeds in this regime; the SQuAD mean of ZO-RGD uses only the three completed runs. These completion rates describe the tested configurations. In the matched rank-2 comparison (Table 14), both variants completed all three seeds on both tasks.

text protocol are disclosed: the LOZO runs of this earlier core used rank 1 with resampling interval $\nu = 50$ (an internal setting, verified from the run logs; the official $\nu = 100$ is used in every unified-protocol table), and the report seeds are $\{0, \ldots, 4\}$ rather than $\{1, 2, 3\}$. Table 16 reports the cells and the paired comparisons. Divergences stay in the denominator: ZO-RGD diverged (FP16 NaN) on SQuAD seeds 0 and 3 and reports 3/5; its RTE cell reports 4/5 after one timeout, and all other cells report 5/5. Figure 2 complements these endpoint cells with a forward-pass-efficiency view of the same regime: RTE development loss against cumulative forward passes at the shared learning rate, truncated at the common 40K-forward budget, with HiZOO added at the same rate.

D.10 Learning-rate diagnostic for the modern-model panel

Table 4 in Section 6.4 tests whether the fixed-rank update formulation transfers beyond OPT under one protocol and three report seeds. Each method selects its own learning rate in a development-only pilot of 2,000 steps. On Llama-3.1-8B BoolQ the pilot chose 10^{-7} for ZO-RGD-S and 10^{-6} for ZO-RGD and LOZO. Table 17 compares both rates at seed 0 for all three methods over the full 20,000-update budget. Changing the rate from 10^{-6} to 10^{-7} increases accuracy by 17.9, 19.8, and 18.2 points for LOZO, ZO-RGD, and ZO-RGD-S, respectively. At the shared rate 10^{-7}, both Riemannian variants reach 85.5, versus LOZO's 80.9; this remaining 4.6-point gap is a single-seed observation with method-specific ranks. The diagnostic identifies learning-rate sensitivity in the panel and shows that the short pilot selected a rate for ZO-RGD and LOZO that performed worse at the full budget on this task. It does not establish a ranking across seeds. The original three-seed panel remains unchanged; its per-seed values are in Appendix D.7.

Table 17: Llama-3.1-8B BoolQ learning-rate diagnostic (accuracy, %). All entries use seed 0 and 20,000 updates; each method retains its panel rank.

Method	Rank	LR 10^{-6}	LR 10^{-7}
LOZO	1	63.0	80.9
ZO-RGD	4	65.7	85.5
ZO-RGD-S	2	67.3	85.5

Table 18: Baselines at ZO-RGD's working point on OPT-1.3B (%). MeZO and LOZO report mean $\pm$ sd over three seeds; ZO-RGD is the single fresh-seed breadth result in Appendix D.4. This diagnostic uses a different working point from the parameterization comparison in Table 5.

Baselines at ZO-RGD's working point	SST-2	RTE	SQuAD
MeZO @ ZO-RGD working point	89.3 ± 0.6	62.9 ± 4.2	72.9 ± 0.8
LOZO @ ZO-RGD working point	89.6 ± 1.3	63.8 ± 1.4	73.1 ± 0.9
ZO-RGD (same point; single seed)	92.4	65.3	77.7

D.11 Additional geometry controls and negative results

Table 5 in Section 6.5 gives the complete finite-difference-geometry and parameterization controls. This appendix adds a working-point diagnostic for the baselines, the traversal-order check, the HiZOO released-implementation diagnostic, and negative results for additional variants.

Baselines at ZO-RGD's working point. Table 18 reruns MeZO and LOZO at ZO-RGD's working point. Its single-seed ZO-RGD reference is descriptive; it is not a three-seed paired comparison.

Block traversal order. Rerunning ZO-RGD-S under three different blockwise traversal orders (name-keyed RNG, three paired seeds) spreads SST-2 accuracy by 0.15 points (91.63/91.78/91.70), far below the seed standard deviation — the fixed traversal order that its scalar recursion follows is not a hidden tuning axis.

LOZO accumulated-update rank. The fixed-rank constraint is a real restriction on the update only because the closest baseline's persistent update is not rank-limited in practice. Instrumenting LOZO's accumulated update $\boldsymbol{W}_T - \boldsymbol{W}_0$ with per-layer SVD sidecars (official $r = 1$ configuration, RTE) shows its effective rank at 90% of the energy reaching a median of 1019 of the 2048 model dimension across 146 tracked OPT-1.3B layers, and 2466 of 5120 across 242 OPT-13B layers, already by step 4,000 and essentially flat thereafter (1019 and 2466 at step 20,000). The baseline's learned update is effectively half full-rank; ours is exactly rank r.

HiZOO released-implementation diagnostic. We include HiZOO (Zhao et al., 2024b) in the OPT-13B RTE comparison and convergence plot using the released implementation at the shared learning rate 10^{-7}. At its published OPT learning rate, however, the same implementation scores at chance on our RTE setup at both 1.3B and 13B (52.1 ± 3.8 / 52.5 ± 1.1). Instrumentation shows that its Hessian-information coefficient is inactive ($\alpha = 0$) in the released setting, at which point the update rule reduces to MeZO's; matched-learning-rate comparisons overlap with MeZO within seed noise at both scales. A modified activated configuration ($\alpha > 0$, FP32 Hessian state) trains in smoke tests, but it is not the published configuration and is therefore not substituted for the released method in the main comparison.

Negative results on our own variants. Two internal variants underperform and are reported as such. A tangent-space (matrix) momentum variant loses to the scalar recursion of ZO-RGD-S on four of six tasks at three paired seeds (pilot: one of six, mean -2.0 points), and a per-step EMA variant wins on one of three tasks; the scalar recursion along the fixed traversal order remains the recommended smoothing. A lazy-retraction variant (retract every F steps, $F \in \{10, 50, 200\}$) is *slower* than retracting every step by 17–19%: phase profiling shows retraction is only 18.4% of a step while random-matrix generation and projection account for $\approx 60\%$, and deferring retraction forces materialization of the accumulated update. Lazy retraction is a separate algorithm outside the scope of the base theorem, and its measured effect is negative on both axes; runtime engineering should target the random-matrix generation and projection phases instead.

Table 19: Normalization ablation on OPT-1.3B. The normalized reference reproduces the historical ZO-RGD $r = 4$ row (Table 9). SQuAD reports F1; the other tasks report accuracy. † The SQuAD unnormalized run diverged and was cancelled before completion; intermediate checkpoints at steps 4K, 8K, and 12K all gave F1 0, and the entry is a truncated run reported at its collapsed value, not a completed evaluation. A rerun cannot repair it: with normalization off, the SQuAD ablation diverges deterministically (first evaluation NaN on all seeds; Appendix D.18), and the divergence itself is the ablation conclusion.

Task	Setting	Unnormalized	Normalized ref.
SST-2	raw, $\epsilon = 10^{-3}$	60.89	90.83
RTE	raw, $\epsilon = 10^{-3}$	55.23	59.57
SQuAD (F1)†	raw, $\epsilon = 10^{-3}$	0.00	76.61
SST-2	raw, $\epsilon = 1.6{\times}10^{-2}$	56.88	90.83

D.12 Effect of normalizing the tangent random matrix

We test the central design choice of the estimator: post-projection normalization that restores the ambient finite-difference radius after tangent projection. Without normalization, the raw projected random matrix has Frobenius norm concentrated at $\sqrt{d_r}$ rather than $\sqrt{mn}$. For OPT-1.3B square blocks with $m = n = 2048$ and $r = 4$, the ratio is $\sqrt{d_r/(mn)} \approx 0.063$, so the unnormalized two-point finite difference evaluates the loss within an effective radius of about 6% of the standard MeZO/LOZO scale at the same ϵ.

Table 19 reports controlled OPT-1.3B ablations with ZO-RGD ($r = 4$, $\eta = 10^{-7}$, batch size 16, 20K steps). The first comparison uses the raw projected random matrix at $\epsilon = 10^{-3}$ on SST-2, RTE, and SQuAD; the rescaled-radius comparison uses $\epsilon = 1.6 \times 10^{-2}$, the square-block compensation value, to test whether one global radius can replace per-block normalization.

The raw projected random matrix performs poorly under the standard $\epsilon = 10^{-3}$ setting on all tested tasks: SST-2 drops by roughly 30 points, RTE stays near chance, and SQuAD yields zero F1 at every measured intermediate evaluation. The $\epsilon = 1.6 \times 10^{-2}$ rescaled-radius comparison does not recover the SST-2 result. This is consistent with the structure of OPT blocks, which mix square attention matrices with non-square feed-forward matrices; a single global ϵ cannot simultaneously calibrate the per-block $\sqrt{d_{r_\ell}/(m_\ell n_\ell)}$ ratios. Normalization is therefore a per-block rank- and shape-dependent calibration of the finite-difference radius, not a global scalar rescaling.

D.13 Rank ablation on OPT-1.3B

Historical convention: the runs in this subsection predate the unified protocol (final-step evaluation, single seed) and support no main-text claim; the protocol-conformant rank evidence is the rank axis of the stability atlas (Appendix D.14).

Rank controls both the intrinsic tangent dimension and the size of the thin-SVD update state. Table 20 swaps the historical-table ranks for ZO-RGD ($r = 4$) and ZO-RGD-S ($r = 2$) on five tasks; the largest observed shift is 1.4 points and most paired results are within 1 point. An SST-2 rank-sweep table that appeared in the earlier version of this manuscript is removed: five of its six cells cannot be reconstructed from any job log; the reconstructible rank evidence under the unified protocol is in the stability atlas. These results indicate that the OPT-1.3B conclusions are not tied to a single isolated rank choice.

D.14 Stability atlas at OPT-1.3B

One-factor-at-a-time robustness of ZO-RGD around the default configuration (SST-2, two paired seeds per setting; Table 21). Rank is insensitive across $r \in \{1, 2, 8\}$ (91.4–92.6 against base 92.6); the perturbation radius is stable one decade down ($\epsilon = 10^{-4}$: 91.3) and breaks one decade up ($\epsilon = 10^{-2}$: 69.8); and the initialization scale is irrelevant across four orders of magnitude ($\sigma_{\text{init}} = 10^{-4}$: 92.6; 10^{-8}: 92.2). In FP16, 10^{-8} underflows to zero, so this configuration effectively starts at $\mathbf{\Sigma} = \mathbf{0}$; training is unaffected because the normalized tangent step does not scale with $\mathbf{\Sigma}$ — the first step has $\|\xi\| \approx 5.5 \times 10^{-3}$ and moves the state off zero, and the step-0 diagnostics of the 10^{-6} and 10^{-8} runs are bitwise identical. The method is therefore insensitive to under-initialization by

Table 20: OPT-1.3B cross-rank auxiliary runs for the two ranks used in the main table. These runs use 20K steps, $\eta = 10^{-7}$, and final-step evaluation; they are separate from the main-table checkpoint-selection convention. SQuAD and DROP report F1; the remaining tasks report accuracy. Bold marks the better rank for each available pair.

Task	Variant	$r = 2$	$r = 4$	Δ
SST-2	ZO-RGD	**91.86**	90.83	-1.03
SST-2	ZO-RGD-S	90.48	**91.86**	$+1.38$
COPA	ZO-RGD	76.00	**77.00**	$+1.00$
COPA	ZO-RGD-S	75.00	75.00	0.00
SQuAD	ZO-RGD-S	77.94	**78.47**	$+0.53$
DROP	ZO-RGD-S	**24.64**	24.35	-0.29

Table 21: Stability atlas (SST-2 accuracy, %; mean $\pm$ sd over two paired seeds; one factor varied per row).

Axis	Arm	SST-2
—	base (r=4, ϵ=10^{-3}, σ_{init}=10^{-6}, FP16)	92.55 ± 0.65
rank	$r = 1$ / $r = 2$ / $r = 8$	91.40 ± 0.49 / 92.60 ± 0.24 / 92.37 ± 0.41
ϵ	10^{-4} / 10^{-2}	91.28 ± 0.00 / 69.84 ± 2.11
init	10^{-4} / 10^{-8}	92.60 ± 0.08 / 92.20 ± 0.65
dtype	bf16	77.64 ± 0.65

Table 22: Manifold-wide $\sigma_{\min}$ and conditioning across the instrumented runs (ranges over completed seeds). All runs start at $\sigma_{\min} = 1.013 \times 10^{-6}$ by construction.

Model / task	Method	$\sigma_{\min}$, step 0	$\sigma_{\min}$, end	max cond.
13B RTE	ZO-RGD	1.013×10^{-6}	0.52–0.60	≤ 1.07
13B RTE	ZO-RGD-S	1.013×10^{-6}	0.63–0.72	≤ 1.07
13B SQuAD	ZO-RGD / -S	1.013×10^{-6}	0.87–1.12	≤ 1.07
30B BoolQ	ZO-RGD / -S	1.013×10^{-6}	1.00–1.15	≤ 1.07

construction. Precision is the one sensitive axis: bf16 scores 77.6 (-14.9 points), and FP16 is the default for this estimator (see also the failure register).

D.15 Singular-value diagnostics

Every 500 steps, each ZO-RGD and ZO-RGD-S run of the shared-learning-rate core (Appendix D.9; 20 runs) and of the two OPT-30B BoolQ configurations (the unified spot-check and the earlier pre-locked triple of Appendix D.6; 4 runs) records the manifold-wide minimum singular value $\sigma_{\min} = \min_\ell \sigma_{r_\ell}(\Delta \boldsymbol{W}_\ell)$, condition-number summaries over the adapted blocks, and one representative layer's spectrum — 24 runs at two scales in total (Table 22; ranges across completed seeds). Three readings: $\sigma_{\min}$ moves *away* from the rank-deficient boundary by five to six orders of magnitude; no singular value approaches zero, so the retraction is never evaluated near the degenerate set; and condition numbers stay within 7% of 1, so no block effectively loses rank by spectral collapse. Of the two ZO-RGD SQuAD seeds that diverge under the shared learning rate (Appendix D.18), seed 0 goes non-finite at step 500, before a second diagnostic is logged; seed 3, which diverges at step 1,500, shows the complementary signature: $\|\boldsymbol{\Sigma}\|$ grows ($0.07 \to 2.58 \to 3.70$ over the diagnostics logged between steps 1,000 and 1,500) while $\sigma_{\min}$ does not collapse before the loss goes non-finite — a step-size and dynamic-range failure, not a boundary failure. The instrument makes the boundary clause of Assumption 1(i) empirically checkable on our own runs; it does not monitor the step-size clause (iv), and it is a diagnostic, not a certificate of the theorem.

D.16 Runtime and memory

Table 6 in Section 6.5 separates forward-pass count, per-step time, and peak memory; it records the computational price of projection and retraction without turning endpoint accuracy into a wall-clock claim. Its forward-only rows average the two hardware repeats in Table 23, which runs all four methods back to back on two different A100-80G cards.

Table 23: Repeated OPT-13B SQuAD cost profile. Each repeat runs all four methods back to back for 1,000 steps each in one allocation, on a different A100-80G card, with intermediate evaluation disabled. These are hardware repeats, not training seeds.

	Time / step (s)		Peak reserved memory (GB)	
Method	Repeat 1	Repeat 2	Repeat 1	Repeat 2
MeZO	2.025	2.030	39.62	39.62
LOZO	1.929	1.931	39.43	39.43
ZO-RGD	3.516	3.508	39.40	39.40
ZO-RGD-S	3.551	3.525	39.38	39.38

D.17 Memory analysis

The persistent trainable state is the thin-SVD triple of Appendix A.4, with $r_\ell(m_\ell+n_\ell+1)$ parameters per selected block. For OPT-1.3B with model dimension 2048, rank $r = 4$, and all 146 selected two-dimensional blocks, the total FP16 fixed-rank update state is approximately

$$\begin{aligned}\sum_\ell r_\ell(m_\ell + n_\ell + 1) \cdot 2 \text{ bytes} \approx \Big(&24[4 \cdot 4(2048 + 2048 + 1) + 2 \cdot 4(2048 + 8192 + 1)] \\ &+ 4(50272 + 2048 + 1) + 4(2050 + 2048 + 1)\Big) \cdot 2 \text{ bytes} \\ &\approx 7.5 \text{ MB}.\end{aligned}$$

The 146 blocks are the matrices updated by the implementation: six projection matrices per transformer layer across 24 OPT layers, plus the token-embedding (50272×2048) and positional-embedding (2050×2048) matrices outside the repeated layer stack. This estimate therefore accounts for the attention projections, MLP projections, and the two embedding matrices in OPT-1.3B. It is negligible relative to the frozen pretrained weights (~ 2.6 GB in FP16). A short empirical measurement (100 training steps, NVIDIA V100, batch 16) reported a 7.2 MB fixed-rank update overhead and a total peak GPU memory of ~ 6.07 GB, dominated by frozen weights and forward-pass activations rather than by the trainable update state.

D.18 Failure register

Every failed run stays in the denominator of its cell or is disclosed here; the manifest retains 63 failed, cancelled, timed-out, or numerically divergent run records alongside 1552 successful result records. The material failures for interpretation are: (i) FP16 divergence of ZO-RGD on OPT-13B SQuAD under the shared learning rate, seeds 0 and 3, first non-finite diagnostic at steps 500 and 1500 respectively, projected-gradient NaN with F1 collapsing to zero, while ZO-RGD-S completed all five seeds in the same regime; one additional ZO-RGD SQuAD attempt in the grid-regime rerun diverged the same way on A100 (from step $\approx$ 12,000) and was superseded by a clean completed rerun. These failures occurred in the listed configurations; both variants completed all runs in the matched rank-2 comparison (Table 14). (ii) bf16: three attempts at a bf16 forensic matrix produced zero completed runs; together with the -14.9-point bf16 atlas entry, FP16 is the supported precision, and a small 2×2 precision forensic shows the divergence onset step shifts with dtype while the mode itself is structural. (iii) The normalization-off ablation diverges deterministically on SQuAD (first evaluation NaN on all seeds); the divergence itself is the ablation conclusion and no rerun can repair it. (iv) OPT-13B ReCoRD, ZO-RGD rank 4 (the single landed sweep cell): manifold-state NaN propagated from step 4,000, and the strict checkpoint loader refused the corrupt state after the 20,000-step run, so the cell produced no valid evaluation and is marked Fail in Table 1; an earlier pre-safeguard attempt of the same configuration collapsed to 15.1. ZO-RGD-S completed the same task at 82.10.