ZO-ACT: EFFICIENT ZEROTH-ORDER FINE-TUNING VIA ONE-SHOT ACTIVATION-INFORMED LOW-RANK SUBSPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Zeroth-order (ZO) optimization enables fine-tuning large language models when backpropagation is unavailable or memory-prohibitive, but existing methods typically perturb the full parameter space or rely on structured low-dimensional perturbations, still leading to noisy gradient estimates and limited optimization flexibility. We propose ZO-Act, an activation-informed ZO fine-tuning method that uses input activations to define a fixed low-rank parameterization. For each linear layer, ZO-Act computes a small activation basis once at initialization and optimizes only lightweight coefficient matrices using forward-only loss evaluations. This explicit coefficient-space parameterization reduces the effective perturbation dimension, enables direct use of standard momentum-based optimizers such as Adam, and naturally supports quantized LLM fine-tuning by keeping low-bit weights frozen. Our analysis characterizes a dimension–alignment trade-off: reducing the perturbation dimension improves ZO gradient estimation, while effective subspace optimization requires preserving sufficient gradient energy. We further show empirically that activation-informed subspaces capture substantially more gradient energy than random and weight-based alternatives. Across Llama-3-8B, OPT-13B, Qwen2.5-7B, Qwen2.5-32B, and INT4 Llama-3-8B, ZO-Act consistently achieves competitive performance against strong ZO baselines across language understanding, question answering, commonsense reasoning, and mathematical reasoning.

1 INTRODUCTION

Fine-tuning large language models (LLMs) (Houlsby et al., 2019; Hu et al., 2021; Gurses et al., 2025; Dettmers et al., 2023) has become a standard technique for adapting pretrained models to downstream tasks. However, as LLMs scale to billions of parameters, conventional first-order fine-tuning becomes increasingly memory-intensive due to backpropagation, activation storage, and optimizer states. Zeroth-order (ZO) optimization provides a promising forward-only alternative, estimating update directions using only loss evaluations and avoiding backpropagation (Malladi et al., 2023). Despite its memory advantage, ZO fine-tuning suffers from high gradient-estimation variance when perturbations are sampled in the full parameter space. MeZO (Malladi et al., 2023) showed that in-place ZO-SGD can fine-tune very large models with inference-level memory, but later studies identified slow convergence and instability as major limitations.

While ZO optimization offers an appealing alternative, matching FO optimization in terms of convergence and accuracy is still challenging. The core issue is that gradients estimated from function queries often have high variance, and this variance worsens in higher-dimensional problems (Duchi et al., 2015; Nesterov & Spokoiny, 2017; Liu et al., 2018). Subsequent work has improved ZO fine-tuning through better optimizers and more structured perturbations, including ZO-Adam and momentum variants (Zhang et al., 2024b), ZO-Muon (Lang et al., 2026), curvature-aware preconditioning (Zhao et al., 2025), low-rank perturbations (Yu et al., 2024; Chen et al., 2025; Lin et al., 2026; Fang et al., 2026), and sparse perturbations (Liu et al., 2026). A particularly effective direction is low-rank or subspace-based ZO perturbation, which reduces the perturbation dimension to improve estimator stability. LOZO (Chen et al., 2025) and SubZero (Yu et al., 2024) restrict perturbations to low-rank matrices to reduce variance. AGZO (Lin et al., 2026) further incorporates activation information into low-rank ZO perturbations, demonstrating the benefit of activation-aware

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.

update directions. ZO-Act leverages activation information in a different way: rather than using it only to guide perturbation construction, we use activations to define a fixed low-dimensional parameterization of the optimization problem. Specifically, each adapted weight update is represented by a frozen activation-informed basis and a small trainable coefficient matrix. ZO optimization is then performed directly over these explicit coefficient variables. This formulation reduces the effective perturbation dimension, enables direct use of standard momentum-based optimizers such as Adam with only lightweight optimizer states, avoids materializing full-weight perturbations, and naturally accommodates quantized models by keeping the original weights frozen.

Building on this formulation, we propose ZO-Act, a one-shot activation-informed ZO fine-tuning method. For each adapted linear layer, ZO-Act computes a low-rank basis from input activations using a single calibration batch, freezes the basis throughout fine-tuning, and optimizes only the corresponding coefficient matrix using forward-only loss evaluations. The resulting fixed parameterization provides a simple and memory-efficient approach to ZO fine-tuning for both full-precision and quantized LLMs.

- We propose ZO-Act, a one-shot activation-informed ZO fine-tuning method that uses input activations to define a fixed low-dimensional parameterization. By freezing the activation basis and optimizing only lightweight coefficient matrices, ZO-Act turns implicit weight-space perturbation into explicit coefficient-space optimization. This reduces the effective perturbation dimension, enables memory-efficient use of standard momentum-based optimizers such as Adam, and naturally supports quantized LLM fine-tuning while keeping the low-bit model frozen.
- We characterize the dimension–alignment trade-off underlying subspace ZO optimization. Reducing the perturbation dimension improves gradient estimation by reducing the dimension-dependent variance and finite-difference error, while effective subspace optimization requires preserving sufficient gradient energy. Empirically, activation-informed subspaces capture substantially more gradient energy than random and weight-based alternatives throughout fine-tuning, supporting both the choice of activation-informed basis and its reuse across optimization steps.
- We demonstrate consistent gains over strong ZO baselines across Llama, OPT, and Qwen models, including Qwen2.5-32B, on language understanding, question answering, commonsense reasoning, and mathematical reasoning. ZO-Act also remains effective with an INT4 Llama-3-8B backbone while preserving the memory advantages of low-bit quantization. Multi-seed evaluations further support the robustness of these gains under comparable or smaller forward-evaluation budgets.

2 Related Work

Zeroth-order fine-tuning. MeZO (Malladi et al., 2023) adapts simultaneous perturbation stochastic approximation to language model fine-tuning by estimating an update direction from two forward losses evaluated under opposite random perturbations. Its in-place implementation avoids storing explicit perturbation vectors and therefore substantially reduces training memory. Nevertheless, MeZO samples perturbations in the full parameter space, so the resulting estimator remains highly noisy in billion-dimensional models.

A number of recent works improve this basic ZO fine-tuning pipeline through optimizer design, curvature information, and structured parameter selection. Zhang et al. (Zhang et al., 2024b) provide a systematic study of ZO variants for LLM adaptation, including momentum, Adam-style updates, conservative updates, block-wise descent, and hybrid ZO-FO training. HiZOO (Zhao et al., 2025) estimates diagonal Hessian information and uses it to precondition ZO updates, addressing the heterogeneous curvature of LLM loss landscapes. ZO-Muon (Lang et al., 2026) combines matrix-structured ZO updates with Muon-style optimization to improve the stability and effectiveness of LLM fine-tuning. These methods demonstrate that reducing estimator noise and improving update geometry are crucial for making ZO fine-tuning competitive.

Structured ZO perturbations. Another closely related line of work reduces the perturbation dimension by imposing structured perturbations. S-MeZO (Liu et al., 2026) reduces the update dimension by selecting a sparse set of sensitive parameters. LOZO (Chen et al., 2025) and SubZero (Yu et al., 2024) perform ZO optimization in randomly generated low-dimensional subspaces, while

LOZO+ (Fang et al., 2026) introduces greedy low-rank subspace selection. ZO-Muon (Lang et al., 2026) further exploits low-rank structure by combining subspace gradient orthogonalization with Muon-style matrix updates, improving the stability and effectiveness of ZO fine-tuning. AGZO (Lin et al., 2026) incorporates activation information into the construction of low-rank ZO perturbations, showing that activation-aware directions can provide more informative update signals than purely random perturbations. ZO-Act is most closely related to this subspace-based line of work, but differs in how the subspace is used. Existing methods primarily use low-dimensional subspaces to construct more effective perturbation directions, whereas ZO-Act uses input activations to define a fixed update parameterization. The activation-informed basis is computed once and kept frozen, and fine-tuning is performed by optimizing only the lightweight coefficient matrices.

3 METHODOLOGY

Motivation. Consider a standard linear layer with input dimension m and output dimension n, $\mathbf{Y} = \mathbf{X}\mathbf{W}$, where $\mathbf{X} \in \mathbb{R}^{b\times m}$, $\mathbf{W} \in \mathbb{R}^{m\times n}$, and $\mathbf{Y} \in \mathbb{R}^{b\times n}$. Let $\mathbf{g}_\mathbf{Y} \in \mathbb{R}^{b\times n}$ denote the gradient of the final loss with respect to the layer output $\mathbf{Y}$. Then the gradient with respect to the weight matrix $\mathbf{W}$ is $\mathbf{g}_\mathbf{W} = \mathbf{X}^\top \mathbf{g}_\mathbf{Y}$. Let $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ be the thin SVD of $\mathbf{X}$, where the diagonal entries of $\mathbf{D}$ are the singular values in descending order. Denote the top-r components by

$$\mathbf{U}_r := \mathbf{U}_{:,:r}, \quad \mathbf{V}_r := \mathbf{V}_{:,:r}, \quad \mathbf{D}_r := \mathbf{D}_{:r,:r}.$$

The weight gradient can be written as

$$\mathbf{g}_\mathbf{W} = \mathbf{V}\mathbf{D}\mathbf{U}^\top \mathbf{g}_\mathbf{Y}.$$

If the singular values of $\mathbf{X}$ decay rapidly, then $\mathbf{g}_\mathbf{W}$ can be well approximated by its rank-r truncation:

$$\mathbf{g}_\mathbf{W} \approx \mathbf{V}_r \mathbf{D}_r \mathbf{U}_r^\top \mathbf{g}_\mathbf{Y}.$$

This suggests that the dominant components of the weight gradient lie in the subspace spanned by the top right singular vectors of the input activation matrix, namely $\mathbf{V}_r$.

The same intuition can also be understood from the perspective of weight perturbation. Suppose we perturb the weight matrix along the activation-informed subspace: $\Delta\mathbf{W} = \mathbf{V}_r\mathbf{R}, \qquad \mathbf{R} \sim \mathcal{N}(0, I)$. Then the induced perturbation on the layer output is

$$\Delta\mathbf{Y} = \mathbf{X}\Delta\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{V}^\top\mathbf{V}_r\mathbf{R} = \mathbf{U}_r\mathbf{D}_r\mathbf{R}.$$

Thus, perturbing $\mathbf{W}$ within the span of $\mathbf{V}_r$ directly targets the dominant activation directions of the layer. In contrast, perturbations outside this subspace are largely suppressed by the small singular values of $\mathbf{X}$ and have limited effect on the layer output. Therefore, restricting zeroth-order perturbations to the $\mathbf{V}_r$ subspace preserves the most effective perturbation directions while substantially reducing the perturbation dimension.

ZO-Act. Motivated by the above observation, we restrict zeroth-order perturbations to the dominant right singular subspace of the input activations. For each linear layer, we first compute a fixed activation-informed basis from a calibration batch, and then perform zeroth-order optimization only over a low-dimensional coefficient matrix.

Consider a linear layer with pretrained weight $\mathbf{W} \in \mathbb{R}^{m\times n}$. Let $\mathbf{V}_r \in \mathbb{R}^{m\times r}$ be the top-r right singular vectors of the input activation matrix $\mathbf{X}$. We parameterize the weight update as $\Delta\mathbf{W} = \mathbf{V}_r\mathbf{B}$, where $\mathbf{B} \in \mathbb{R}^{r\times n}$ is the only trainable parameter for this layer. The effective weight is therefore

$$\mathbf{W}_{\text{eff}} = \mathbf{W} + \mathbf{V}_r\mathbf{B}.$$

The pretrained weight $\mathbf{W}$ and the activation basis $\mathbf{V}_r$ are both frozen throughout training, while $\mathbf{B}$ is initialized as zero.

At each training step, we sample a random perturbation in the low-dimensional coefficient space: $\mathbf{Z} \sim \mathcal{N}(0, I), \mathbf{Z} \in \mathbb{R}^{r\times n}$. The corresponding weight-space perturbation is

$$\Delta\mathbf{W}_{\text{ZO}} = \mathbf{V}_r\mathbf{Z}.$$

Given a perturbation magnitude $\mu > 0$, the forward-perturbed effective weight is

$$\mathbf{W}^+_{\text{eff}} = \mathbf{W} + \mathbf{V}_r(\mathbf{B} + \mu\mathbf{Z}).$$

Algorithm 1 Subspace Initialization

Require: Pretrained model, calibration batch $\mathcal{B}_{\text{cal}}$, rank r

1: Run one forward pass on $\mathcal{B}_{\text{cal}}$ and collect input activations $\mathbf{X}_\ell$ for each linear layer ℓ.
2: **for** each linear layer ℓ with weight $\mathbf{W}_\ell \in \mathbb{R}^{m_\ell \times n_\ell}$ **do**
3: Compute the top-r right singular vectors of $\mathbf{X}_\ell$:

$$\mathbf{X}_\ell \approx \mathbf{U}_{\ell,r}\mathbf{D}_{\ell,r}\mathbf{V}_{\ell,r}^\top.$$

4: Freeze the pretrained weight $\mathbf{W}_\ell$ and the basis $\mathbf{V}_{\ell,r}$.
5: **end for**
6: **return** Frozen bases $\mathbf{V}_{\ell,r}$.

Algorithm 2 ZO-Act

Require: Data $\mathcal{D}$, pretrained weights $\{\mathbf{W}_\ell\}_\ell$, rank r, perturbation scale μ, learning rate η, queries q, steps T

1: Sample calibration batch $\mathcal{B}_{\text{cal}} \subset \mathcal{D}$.
2: Obtain frozen subspaces $\{\mathbf{V}_{\ell,r}\}_\ell$ using Algorithm 1 with $\mathcal{B}_{\text{cal}}$ and rank r.
3: Initialize the trainable matrix $\mathbf{B}_\ell \leftarrow \mathbf{0} \in \mathbb{R}^{r \times n_\ell}$ for all ℓ.
4: **for** $t = 1, \ldots, T$ **do**
5: Sample mini-batch $\mathcal{B}_t \subset \mathcal{D}$.
6: Evaluate $\mathcal{L}$ with $\mathbf{W}_{\ell,\text{eff}} = \mathbf{W}_\ell + \mathbf{V}_{\ell,r}\mathbf{B}_\ell$.
7: Initialize $\widehat{\mathbf{g}}_{\mathbf{B}_\ell} \leftarrow 0$ for all ℓ.
8: **for** $j = 1, \ldots, q$ **do**
9: Sample $\mathbf{Z}_\ell^{(j)} \sim \mathcal{N}(0, I)$ for all ℓ.
10: Evaluate $\mathcal{L}^{(j,+)}$ with $\mathbf{W}_{\ell,\text{eff}}^{(j,+)} = \mathbf{W}_\ell + \mathbf{V}_{\ell,r}(\mathbf{B}_\ell + \mu\mathbf{Z}_\ell^{(j)})$.
11: $a^{(j)} \leftarrow (\mathcal{L}^{(j,+)} - \mathcal{L})/\mu$.
12: $\widehat{\mathbf{g}}_{\mathbf{B}_\ell} \leftarrow \widehat{\mathbf{g}}_{\mathbf{B}_\ell} + a^{(j)}\mathbf{Z}_\ell^{(j)}$ for all ℓ.
13: **end for**
14: $\widehat{\mathbf{g}}_{\mathbf{B}_\ell} \leftarrow \widehat{\mathbf{g}}_{\mathbf{B}_\ell}/q$ for all ℓ.
15: Update each $\mathbf{B}_\ell$ with an optimizer using $\widehat{\mathbf{g}}_{\mathbf{B}_\ell}$.
16: **end for**
17: **return** Fine-tuned model with $\mathbf{W}_{\ell,\text{eff}} = \mathbf{W}_\ell + \mathbf{V}_{\ell,r}\mathbf{B}_\ell$.

For an input activation $\mathbf{X}$, the perturbed forward pass can be written as

$$\mathbf{Y}^+ = \mathbf{X}\mathbf{W} + (\mathbf{X}\mathbf{V}_r)(\mathbf{B} + \mu\mathbf{Z}).$$

Let $\mathcal{L}^+$ denote the loss from the forward-perturbed model and let $\mathcal{L}$ denote the unperturbed loss on the same mini-batch. The forward-difference zeroth-order estimator for $\mathbf{B}$ is

$$\widehat{\mathbf{g}}_{\mathbf{B}} = \frac{\mathcal{L}^+ - \mathcal{L}}{\mu}\mathbf{Z}.$$

With q independent perturbation directions, we average the estimates:

$$\widehat{\mathbf{g}}_{\mathbf{B}} = \frac{1}{q}\sum_{j=1}^{q} \frac{\mathcal{L}^{(j,+)} - \mathcal{L}}{\mu}\mathbf{Z}^{(j)}.$$

We then update $\mathbf{B}$ using a first-order optimizer such as Adam with the estimated gradient $\widehat{\mathbf{g}}_{\mathbf{B}}$. Since the perturbation is sampled in $\mathbb{R}^{r \times n}$ rather than $\mathbb{R}^{m \times n}$, the effective perturbation dimension is reduced from mn to rn, where $r \ll m$. The whole ZO-Act algorithm is summarized in Algorithm 1 and Algorithm 2.

4 Theoretical Analysis

We analyze ZO-Act as zeroth-order optimization over a restricted coefficient space. The analysis highlights a variance–approximation trade-off. Restricting optimization to a low-dimensional coefficient space reduces the dimension-dependent variance and finite-difference error of zeroth-order gradient estimation. At the same time, restricting updates to an activation-informed subspace leaves an out-of-subspace gradient component that cannot be optimized directly. We first characterize the optimization benefit in the coefficient space, and then analyze when the activation-informed subspace preserves a large fraction of the full weight-gradient energy.

Let $F(\theta)$ denote the fine-tuning objective, where $\theta \in \mathbb{R}^d$ collects the weight parameters of all adapted linear layers. ZO-Act freezes the pretrained weights and the activation-informed bases, and optimizes only the coefficient matrices $\{\mathbf{B}_\ell\}_{\ell=1}^L$. Let $\beta \in \mathbb{R}^k$ collect all trainable coefficients, where $k = \sum_{\ell=1}^L rn_\ell$. Equivalently, there exists a fixed activation-informed embedding U such that $\theta(\beta) = \theta_0 + U\beta$, where θ_0 denotes the frozen pretrained weights. Since each activation basis has orthonormal columns, the columns of U can be taken to be orthonormal. Thus, ZO-Act optimizes the restricted objective

$$\phi(\beta) = F(\theta_0 + U\beta)$$

rather than the full objective $F(\theta)$ over all weight entries. We first quantify how the coefficient-space dimension affects the accuracy of ZO gradient estimation. Let $g_t = \nabla\phi(\beta_t)$ and let k denote the number of coefficients. For a small finite-difference radius μ, the Gaussian ZO estimator

$$\widehat{g}_t = \frac{\phi(\beta_t + \mu z_t) - \phi(\beta_t)}{\mu} z_t, \qquad z_t \sim \mathcal{N}(0, I_k),$$

has the first-order approximation $a_t = \langle g_t, z_t\rangle z_t$. Conditional on β_t, this approximation satisfies

$$\mathbb{E}[a_t] = g_t, \qquad \mathbb{E}\|a_t - g_t\|^2 = (k+1)\|g_t\|^2.$$

Averaging q independent directions therefore gives

$$\mathbb{E}\left\|\frac{1}{q}\sum_{j=1}^q a_t^{(j)} - g_t\right\|^2 = \frac{k+1}{q}\|g_t\|^2.$$

Thus, to leading order, the relative estimation variance scales as $(k+1)/q$, motivating ZO estimation in a low-dimensional coefficient space. This variance reduction translates into a convergence guarantee of ZO-Act under gradient descent, as stated in the following theorem.

Theorem 1 (Informal)**.** *Let $\phi(\beta) = F(\theta_0 + U\beta)$, where $\beta \in \mathbb{R}^k$ collects all trainable coefficient matrices and U has orthonormal columns. Assume that F is L_F-smooth and that ϕ is lower bounded by $\phi_k^{\inf}$. When ZO-Act applies gradient descent using a q-query one-sided Gaussian zeroth-order estimator with perturbation scale μ, the iterates satisfy*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\phi(\beta_t)\|^2 \le O\left(\frac{L_F\left(1+\frac{k}{q}\right)\left(\phi(\beta_0) - \phi_k^{\inf}\right)}{T}\right) + O\left(L_F^2\mu^2k^3\right).$$

The proof is given in Appendix A.1. Theorem 1 shows the benefit of optimizing in a low-dimensional coefficient space: reducing k improves the dimension-dependent convergence factor and reduces the finite-difference error bound. Full-weight ZO operates in dimension $d = \sum_{\ell=1}^L m_\ell n_\ell$, whereas ZO-Act uses only $k = \sum_{\ell=1}^L rn_\ell \ll d$.

The effectiveness of this reduction also depends on the selected subspace. Let $\theta_t = \theta_0 + U\beta_t$ and $P_U = UU^\top$. Since U has orthonormal columns,

$$\|\nabla F(\theta_t)\|^2 = \|\nabla\phi(\beta_t)\|^2 + \|(I - P_U)\nabla F(\theta_t)\|^2.$$

Combining this identity with Theorem 1 gives

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\theta_t)\|^2 \le O\left(\frac{L_F(1+k/q)\left(\phi(\beta_0) - \phi_k^{\inf}\right)}{T}\right) + O\left(L_F^2\mu^2k^3\right) + \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|(I - P_U)\nabla F(\theta_t)\|^2.$$

The final term is the gradient energy outside the update subspace. Thus, an effective subspace should keep the coefficient dimension small while preserving as much gradient energy as possible. To quantify this alignment, consider a linear layer with weight gradient

$$\mathbf{G} = \mathbf{X}^\top\mathbf{D}, \qquad \mathbf{D} = \partial\mathcal{L}/\partial\mathbf{Y}.$$

For an orthonormal rank-r basis $\mathbf{V}$, define the gradient capture ratio

$$\rho(\mathbf{V}) = \frac{\|\mathbf{V}\mathbf{V}^\top\mathbf{G}\|_F^2}{\|\mathbf{G}\|_F^2}.$$

The corresponding residual is

$$\|(I - \mathbf{V}\mathbf{V}^\top)\mathbf{G}\|_F^2 = \big(1 - \rho(\mathbf{V})\big)\|\mathbf{G}\|_F^2,$$

so greater capture directly reduces the out-of-subspace gradient energy.

The activation spectrum helps explain why ZO-Act can achieve high capture. Writing the compact SVD as $\mathbf{X} = \sum_{i=1}^{s} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$, with $\sigma_1 \geq \cdots \geq \sigma_s > 0$, gives

$$\mathbf{G} = \sum_{i=1}^{s} \sigma_i \mathbf{v}_i (\mathbf{D}^\top \mathbf{u}_i)^\top.$$

Thus, for the activation basis $\mathbf{V}_r = [v_1, \ldots, v_r]$,

$$\rho(\mathbf{V}_r) = \frac{\sum_{i=1}^{r} \sigma_i^2 \|\mathbf{D}^\top \mathbf{u}_i\|^2}{\sum_{i=1}^{s} \sigma_i^2 \|\mathbf{D}^\top \mathbf{u}_i\|^2}.$$

Each direction's contribution is weighted by both its activation singular value and the corresponding backward signal. When the activation spectrum is concentrated in its leading directions, these directions can retain much of the gradient energy, provided that the backward signal does not disproportionately favor the trailing directions. Under this condition, even a small activation subspace can leave only a modest out-of-subspace residual.

Our measurements show that the dominant activation directions capture substantially more gradient energy than dimension-matched random subspaces and achieve better fine-tuning performance, as shown in Figure 1, Section 5.4, and Section 5.5. This supports ZO-Act's choice of a subspace that preserves useful gradient directions while keeping the coefficient space small.

Taken together, the analysis reveals a dimension–alignment trade-off in subspace ZO optimization. A smaller coefficient space improves gradient estimation by reducing the dimension-dependent variance and finite-difference error, but is useful only if it preserves sufficient gradient energy. ZO-Act is designed to balance these two factors: the activation-informed basis provides a highly compact update space while retaining directions that are empirically well aligned with gradients.

5 Experiments

We evaluate ZO-Act on forward-only fine-tuning tasks for large language models. Our experiments cover language understanding and question answering, commonsense reasoning, mathematical reasoning, and quantized LLM fine-tuning. For full-precision models, we consider Llama-3-8B (Grattafiori et al., 2024),OPT-13B (Zhang et al., 2022), Qwen2.5-7B, and QWen2.5-32B (Hui et al., 2024). For quantized fine-tuning, we evaluate an INT4 quantized Llama-3-8B model, denoted as Llama-3-8B-w4, which is quantized by MagR (Zhang et al., 2024a).

To ensure a fair comparison with existing ZO baselines, we match or slightly reduce the total number of forward passes used by ZO-Act compared with the corresponding baseline methods whenever possible. Unless otherwise specified, ZO-Act uses Gaussian perturbations with $q = 8$ perturbation directions per update step and perturbation magnitude $\mu = 10^{-3}$. ZO-Act uses rank $r = 1$. We use the forward-difference estimator and optimize the low-rank coefficient matrices with Adam. Full details are shown in Appendix B.

5.1 Language Understanding

Models and Datasets. Following the setting of Lang et al. (2026), we evaluate ZO-Act on six standard language understanding and question answering benchmarks: SST-2, RTE, CB, BoolQ, WiC, and SQuAD. We report F1 score for SQuAD and accuracy for the remaining tasks. Experiments are conducted on two full-precision model families, Llama-3-8B and OPT-13B. We additionally report results on an INT4 quantized Llama-3-8B model.

Results. Table 1 summarizes the language understanding and question answering results. ZO-Act with Adam achieves strong performance across both model families, with particularly clear gains over prior ZO methods on RTE and CB for Llama-3-8B and on SST-2, RTE, and BoolQ for OPT-13B. On the remaining tasks, its performance is generally competitive with the strongest reported ZO results. ZO-Act with SGD already achieves competitive performance, while Adam provides

Method	Llama-3-8B						OPT-13B					
	SST-2	RTE	CB	BoolQ	WiC	SQuAD	SST-2	RTE	CB	BoolQ	WiC	SQuAD
• Adam (FO)	96.0	92.0	92.0	86.6	72.6	90.4	95.3	80.9	94.6	83.5	66.3	89.5
• LoRA (FO)	95.0	80.9	73.2	86.4	70.7	89.4	94.8	78.3	69.6	80.2	64.3	88.0
MeZO	92.7	74.4	69.6	76.7	57.8	86.7	91.4	66.1	66.0	66.1	59.4	81.8
S-MeZO	92.1	69.7	69.6	80.5	56.9	87.5	90.4	63.5	69.6	66.4	58.8	80.8
HiZOO	93.5	75.1	69.6	80.0	59.7	87.3	92.1	69.3	69.6	67.6	59.4	82.1
LOZO	92.5	66.8	69.6	79.4	55.8	89.0	91.7	70.4	69.6	71.9	60.2	84.9
LOZO+	93.5	76.5	75.0	83.1	63.2	88.5	92.7	66.8	**71.4**	71.3	59.3	84.2
SubZero	92.1	71.4	67.9	82.0	58.8	88.3	92.1	71.8	**71.4**	70.8	60.8	84.5
Subspace-MeZO	92.3	68.6	69.6	80.0	62.9	84.5	91.7	70.7	**71.4**	68.1	**61.7**	83.5
ZO-Muon	94.3	81.2	69.6	82.9	65.2	88.2	92.5	72.9	**71.4**	72.4	**61.7**	84.5
AGZO	93.4	83.0	71.4	84.5	62.4	**90.3**	90.5	67.5	66.0	68.1	56.0	**85.4**
ZO-Act Full Adam	**94.6**	**87.0**	82.1	**85.1**	**69.1**	89.1	**94.0**	**74.0**	69.6	**73.1**	59.4	85.1
ZO-Act Full SGD	94.0	84.5	78.6	83.9	64.7	89.5	92.4	67.9	69.6	69.9	58.2	83.9
ZO-Act INT4 Adam	93.4	83.4	**85.7**	83.8	62.1	87.9	–	–	–	–	–	–

Table 1: Language understanding and question answering results on Llama-3-8B and OPT-13B. SST-2, RTE, CB, BoolQ, and WiC report accuracy, while SQuAD reports F1.

Method	ARC-c	ARC-e	BoolQ	HellaS	OBQA	PIQA	SIQA	WinoG	Average
Zero-shot	10.0	12.5	61.9	12.3	23.4	47.4	2.2	0.0	21.2
LOZO	51.9	69.6	62.6	48.3	53.0	70.2	54.5	50.6	57.6
LOZO+	59.9	79.0	62.2	68.0	54.4	77.2	59.4	52.9	64.1
AGZO	65.0	83.3	64.5	70.7	60.8	**79.9**	63.5	57.7	68.2
SubZero	60.2	78.3	62.5	68.0	53.6	76.7	59.2	52.6	63.9
HiZOO	60.3	79.6	63.6	65.6	58.2	73.1	58.6	53.5	64.1
ZO-Muon	61.4	78.0	63.0	63.6	50.8	76.7	57.3	53.6	63.1
ZO-Act	**68.7**	**85.9**	**64.7**	**76.4**	**65.6**	79.6	**67.3**	**58.4**	**70.8**
ZO-Act INT4	60.1	79.8	60.9	71.6	60.8	77.6	62.7	54.2	66.0

Table 2: Commonsense reasoning results on Llama-3-8B. Average is the mean over the eight tasks.

further improvements on most tasks. This highlights an additional benefit of the explicit coefficient-space parameterization: standard adaptive optimizers can be directly applied to the low-dimensional trainable variables. Unlike full-parameter ZO, where Adam requires optimizer states for the entire model, ZO-Act maintains these states only for its lightweight coefficient matrices, making adaptive optimization memory-efficient. The gains suggest that ZO-Act can benefit from momentum and adaptive scaling without sacrificing the memory efficiency of forward-only fine-tuning. These results support the effectiveness of the activation-informed coefficient space across tasks and model families. Additional multi-seed results on Llama-3-8B and evaluations on Qwen2.5-7B and Qwen2.5-32B are provided in Appendix A.3 and Appendix A.4, respectively.

ZO-Act also remains effective when applied to the INT4 Llama-3-8B model. Despite the reduced weight precision, the quantized model remains competitive with strong full-precision ZO baselines across the evaluated tasks. This demonstrates that ZO-Act can retain strong adaptation performance while operating with a low-bit backbone. We further examine this quantized setting and its practical advantages in Section 5.3.

5.2 Open-Ended Generation Tasks

Models and Datasets. We further evaluate ZO-Act on both commonsense and mathematical reasoning. For commonsense reasoning, we consider eight tasks: ARC-Challenge, ARC-Easy (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), and WinoGrande (Sakaguchi et al., 2021). We fine-tune Llama-3-8B on the combined training data and report accuracy on each task individually, together with the average accuracy over the eight tasks. All commonsense tasks are evaluated using open-ended generation with answer-match scoring. To further evaluate ZO fine-tuning on more structured multi-step reasoning, we consider GSM8K (Cobbe et al., 2021), which requires

Method	Forward	Seed				Mean ± Std
		0	13	42	100	
ZO-Muon	90k	41.6	34.4	42.7	35.6	38.6 ± 4.2
LOZO+	100k	41.7	38.0	40.0	40.6	40.1 ± 1.6
AGZO	100k	42.2	43.4	41.8	37.6	41.3 ± 2.5
ZO-Act	90k	**44.5**	**44.9**	**45.6**	**46.4**	**45.3 ± 0.8**

Table 3: GSM8K on Llama-3-8B across four seeds.

the model to generate reasoning steps before producing the final numerical answer. It therefore provides a complementary evaluation of ZO-Act on open-ended mathematical reasoning.

Results. Table 2 reports the commonsense reasoning results. ZO-Act achieves the best average performance among all evaluated ZO methods, reaching 70.8 average accuracy, compared with 68.2 for the strongest baseline, AGZO. ZO-Act obtains the best result on seven of the eight individual tasks and is within 0.3 points of the best result on PIQA. The improvements are particularly pronounced on ARC-Challenge, HellaSwag, OpenBookQA, and SIQA, showing that the activation-informed coefficient space remains effective across a diverse collection of reasoning tasks. ZO-Act also remains effective when applied to the INT4 backbone, achieving an average accuracy of 66.0 despite the reduced weight precision. Table 3 further evaluates ZO-Act on GSM8K across four random seeds. ZO-Act consistently outperforms the evaluated ZO baselines across different seeds while also exhibiting substantially lower run-to-run variation. This demonstrates that its improvement on mathematical reasoning is both stronger and more stable, rather than being driven by a favorable random seed. Notably, ZO-Act achieves this performance with fewer optimization steps than several competing methods. Together with the commonsense results, these experiments demonstrate that ZO-Act generalizes beyond language understanding tasks to open-ended, multi-step reasoning while maintaining stable optimization across runs.

5.3 ZO-Act on Quantized Models

ZO-Act naturally extends to quantized LLMs. Most existing ZO methods perturb and update the original model weights, which is not well aligned with low-bit quantized models. Continuous perturbations are difficult to apply directly to discrete quantized weights, while maintaining a full-precision shadow copy would largely offset the memory benefit of quantization and ZO fine-tuning. In contrast, ZO-Act keeps the quantized backbone frozen and performs both optimization and perturbation only in the small full-precision coefficient space. Thus, no full-precision copy of the backbone is required.

Empirically, ZO-Act remains competitive with strong full-precision ZO baselines under INT4 quantization on both language understanding and generation-based reasoning tasks, as shown in Tables 1 and 2. These results demonstrate that ZO-Act can retain strong adaptation performance while preserving the memory advantages of a low-bit quantized model.

Subspace initialization	Accuracy (%)
ZO-Act	**87.0**
Top-r weight SVD	64.3
Bottom-r weight SVD	59.6
Random	63.9
Random orthogonal	52.0

Table 4: Effect of subspace initialization on RTE with Llama-3-8B at $r = 1$.

5.4 Ablation on Subspace Selection

We examine the role of subspace selection by varying only the initialization of V, keeping all other settings fixed. Experiments use Llama-3-8B on RTE with $r = 1$, $q = 8$, and the Adam optimizer. We compare ZO-Act's dominant activation subspace with subspaces constructed from the top or bottom right singular vectors of the weight matrix, as well as random Gaussian and random orthogonal bases.

Table 4 shows that ZO-Act substantially outperforms all alternative initializations. In particular, simply restricting optimization to a rank-1 subspace is insufficient. The improvement therefore comes not only from dimensionality reduction, but also from selecting an informative subspace. This supports the central motivation of ZO-Act and is consistent with the gradient-alignment analysis in Section 5.5.

5.5 Persistent Gradient Alignment of Activation Subspaces

ZO-Act constructs an activation-informed basis at initialization and keeps it fixed throughout fine-tuning. To assess whether this basis remains informative, we track the captured gradient energy during a 300-step first-order full fine-tuning run. Let $\mathbf{V}_{\ell,0}$ denote the initial top-r right singular vectors of the input activation matrix for layer ℓ. We measure gradient alignment by

$$\rho_{\ell,t} = \frac{\|\mathbf{V}_{\ell,0}^{\top}\mathbf{G}_{\ell,t}\|_F^2}{\|\mathbf{G}_{\ell,t}\|_F^2},$$

averaged over all target linear layers.

As shown in Figure 1, the fixed activation subspace captures substantially more gradient energy than random and weight-based subspaces throughout training. Capture largely stabilizes after approximately 100 steps, supporting the continued use of the initial basis. Even a single activation direction retains a good fraction of the gradient energy. Increasing the rank provides broader coverage but also enlarges the coefficient space in which the zeroth-order gradient is estimated. The similar accuracy across ranks (Section A.2) is consistent with this trade-off, suggesting that a small activation subspace can preserve useful update directions while keeping gradient estimation low-dimensional. Appendix A.5 further shows that the same qualitative ordering also holds along the actual ZO-Act training trajectory.

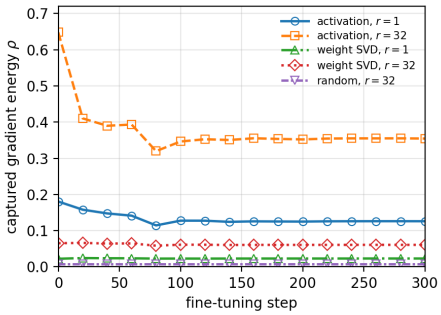


Figure 1: Gradient energy captured by the fixed activation subspace during fine-tuning of Llama-3-8B on RTE.

5.6 Efficiency Comparison.

We compare the practical efficiency of ZO-Act with representative ZO fine-tuning baselines in Table 5. The comparison is conducted on the RTE task using Llama-3-8B, and all methods are evaluated under the same total forward-query budget of 4500 forward passes.

Method	Runtime (min)	Memory (GB)
LOZO	45	**16.1**
LOZO+	**42**	16.3
SubZero	46	**16.1**
AGZO	47	**16.1**
ZO-Muon	43	19.0
HiZOO	94	30.0
ZO-Act	**42**	**16.1**

Table 5: Runtime and memory on Llama-3-8B.

ZO-Act achieves the fastest runtime among the evaluated methods while matching the lowest memory usage. Its memory is close to LOZO, LOZO+, SubZero, and AGZO, and lower than ZO-Muon and HiZOO. The one-shot subspace initialization adds negligible overhead, taking only **1.5** seconds before fine-tuning. This indicates that the activation-informed coefficient-space design introduces little practical overhead while retaining the memory efficiency of forward-only fine-tuning. Moreover, when ZO-Act is applied to the INT4 quantized model, the memory usage is further reduced to **5.8GB**, demonstrating its advantage for low-bit fine-tuning.

6 Conclusion

We proposed ZO-Act, a one-shot activation-informed zeroth-order fine-tuning method that uses input activations to define a fixed low-dimensional parameterization and optimizes only lightweight coefficient matrices. This design reduces the effective perturbation dimension, enables memory-efficient use of standard momentum-based optimizers, and naturally supports quantized LLMs with frozen low-bit models. Our analysis characterizes the dimension–alignment trade-off underlying subspace ZO optimization, while experiments show that activation-informed subspaces preserve useful gradient directions and consistently improve over strong ZO baselines across model families and downstream tasks. These results establish activation-informed coefficient-space optimization as an effective and practical approach to forward-only LLM fine-tuning.

7 AI USE STATEMENT

We used AI tools to assist with algorithm implementation, experiments, and manuscript revision for clarity and readability. The authors reviewed all AI-assisted contributions and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

8 REPRODUCIBILITY STATEMENT

All experiments use publicly available datasets. Detailed experimental settings, implementation details, and hyperparameter configurations are provided in Appendix B for reproducibility.

REFERENCES

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Yiming Chen, Yuan Zhang, Liyuan Cao, Kun Yuan, and Zaiwen Wen. Enhancing zeroth-order fine-tuning for language models with low-rank structures. In *International Conference on Learning Representations*, volume 2025, pp. 62581–62607, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. 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. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 9, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

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

Jinjie Fang, Chengxun Jin, Tianxing Man, Yi Chang, and Bin Gu. Lozo+: Provably efficient zeroth-order fine-tuning via greedy low-rank subspace selection. In *Forty-third International Conference on Machine Learning*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Selcuk Gurses, Aozhong Zhang, Yanxia Deng, Xun Dong, Xin Li, Naigang Wang, Penghang Yin, and Zi Yang. Diablo: Diagonal blocks are sufficient for finetuning. *arXiv preprint arXiv:2506.03230*, 2025.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. URL `https://api.semanticscholar.org/CorpusID:235458009`.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Yicheng Lang, Changsheng Wang, Yihua Zhang, Mingyi Hong, Zheng Zhang, Wotao Yin, and Sijia Liu. Powering up zeroth-order training via subspace gradient orthogonalization. *arXiv preprint arXiv:2602.17155*, 2026.

Wei Lin, Yining Jiang, Qingyu Song, Qiao Xiang, and Hong Xu. Agzo: Activation-guided zeroth-order optimization for llm fine-tuning. *arXiv preprint arXiv:2601.17261*, 2026.

Sijia Liu, Jie Chen, Pin-Yu Chen, and Alfred Hero. Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications. In *International Conference on Artificial Intelligence and Statistics*, pp. 288–297. PMLR, 2018.

Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Sparse mezo: Less parameters for better performance in zeroth-order llm fine-tuning. *Advances in Neural Information Processing Systems*, 38:21139–21169, 2026.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Ziming Yu, Pan Zhou, Sike Wang, Jia Li, and Hua Huang. Subzero: Random subspace zeroth-order optimization for memory-efficient llm fine-tuning, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Aozhong Zhang, Naigang Wang, Yanxia Deng, Xin Li, Zi Yang, and Penghang Yin. Magr: Weight magnitude reduction for enhancing post-training quantization. *Advances in neural information processing systems*, 37:85109–85130, 2024a.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

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

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

A Proofs and Analysis

A.1 Formal Statement and Proof of Theorem 1

Theorem 2 (Convergence of ZO-Act)**.** *Let $\phi(\beta) = F(\theta_0 + U\beta)$, where $\beta \in \mathbb{R}^k$ collects all trainable coefficient matrices. Assume that F is L_F-smooth, $U^\top U = I$, and ϕ is lower bounded by $\phi_k^{\inf}$. At iteration t, ZO-Act samples q independent Gaussian directions $z_t^{(j)} \sim \mathcal{N}(0, I_k)$ and uses the one-sided estimator*

$$\widehat{g}_t = \frac{1}{q}\sum_{j=1}^{q} \frac{\phi(\beta_t + \mu z_t^{(j)}) - \phi(\beta_t)}{\mu} z_t^{(j)}.$$

Suppose the update is $\beta_{t+1} = \beta_t - \eta \widehat{g}_t$. If

$$0 < \eta \le \frac{1}{8L_F\left(1 + \frac{k+1}{q}\right)},$$

then

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla\phi(\beta_t)\|^2 \le \frac{4\left(\phi(\beta_0) - \phi_k^{\inf}\right)}{\eta T} + \frac{5}{8}L_F^2\mu^2 k(k+2)(k+4).$$

In particular, choosing $\eta = \frac{1}{8L_F\left(1+\frac{k+1}{q}\right)}$ gives

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla\phi(\beta_t)\|^2 \le \frac{32L_F\left(1 + \frac{k+1}{q}\right)\left(\phi(\beta_0) - \phi_k^{\inf}\right)}{T} + \frac{5}{8}L_F^2\mu^2 k(k+2)(k+4).$$

Proof. Since F is L_F-smooth and $U^\top U = I$, the restricted objective $\phi(\beta) = F(\theta_0 + U\beta)$ is also L_F-smooth. Indeed, $\nabla\phi(\beta) = U^\top \nabla F(\theta_0 + U\beta)$, and hence

$$\|\nabla\phi(\beta) - \nabla\phi(\beta')\| \le L_F\|\beta - \beta'\|.$$

Let $g_t = \nabla\phi(\beta_t)$. For each Gaussian direction $z_t^{(j)}$, define

$$\widehat{g}_t^{(j)} = \frac{\phi(\beta_t + \mu z_t^{(j)}) - \phi(\beta_t)}{\mu} z_t^{(j)},$$

$$\widehat{g}_t = \frac{1}{q}\sum_{j=1}^{q} \widehat{g}_t^{(j)}.$$

By L_F-smoothness of ϕ, for any z,

$$\phi(\beta_t + \mu z) = \phi(\beta_t) + \mu\langle g_t, z\rangle + R_t(z),$$

$$|R_t(z)| \le \frac{L_F\mu^2}{2}\|z\|^2.$$

We write $\widehat{g}_t^{(j)} = a_t^{(j)} + b_t^{(j)}$ where, $a_t^{(j)} = \langle g_t, z_t^{(j)}\rangle z_t^{(j)}$, and $b_t^{(j)} = \frac{R_t(z_t^{(j)})}{\mu} z_t^{(j)}$.

The remainder term satisfies $\|b_t^{(j)}\| \le (L_F\mu/2)\|z_t^{(j)}\|^3$. Let $\bar{a}_t = q^{-1}\sum_{j=1}^q a_t^{(j)}$ and $\bar{b}_t = q^{-1}\sum_{j=1}^q b_t^{(j)}$, so that $\widehat{g}_t = \bar{a}_t + \bar{b}_t$.

We first lower bound the expected descent direction. Since $z_t^{(j)} \sim \mathcal{N}(0, I_k)$, we have $\mathbb{E}[a_t^{(j)}] = g_t$, and hence $\mathbb{E}[\bar{a}_t] = g_t$. Moreover,

$$\mathbb{E}\|\bar{b}_t\| \le \frac{1}{q}\sum_{j=1}^{q}\mathbb{E}\|b_t^{(j)}\| \le \frac{L_F\mu}{2}\mathbb{E}\|z\|^3.$$

It holds $\mathbb{E}\|z\|^3 \le (\mathbb{E}\|z\|^6)^{1/2}$ and $\mathbb{E}\|z\|^6 = k(k+2)(k+4)$. We define $M_k = k(k+2)(k+4)$, Then $\mathbb{E}\|\bar{b}_t\| \le (L_F\mu/2)\sqrt{M_k}$. Thus,

$$\begin{aligned}\mathbb{E}\langle g_t, \widehat{g}_t\rangle &= \mathbb{E}\langle g_t, \bar{a}_t\rangle + \mathbb{E}\langle g_t, \bar{b}_t\rangle \\ &\ge \|g_t\|^2 - \|g_t\|\mathbb{E}\|\bar{b}_t\| \\ &\ge \|g_t\|^2 - \left(\frac{1}{2}\|g_t\|^2 + \frac{1}{2}(\mathbb{E}\|\bar{b}_t\|)^2\right) \\ &= \frac{1}{2}\|g_t\|^2 - \frac{L_F^2\mu^2}{8}M_k.\end{aligned}$$

Next, we upper bound the second moment of $\widehat{g}_t$. For the leading Gaussian term,

$$\mathbb{E}\|a_t^{(j)}\|^2 = \mathbb{E}[\langle g_t, z\rangle^2\|z\|^2] = (k+2)\|g_t\|^2.$$

Therefore, since the $a_t^{(j)}$'s are independent and each has mean g_t,

$$\begin{aligned}\mathbb{E}\|\bar{a}_t\|^2 &= \|\mathbb{E}\bar{a}_t\|^2 + \mathbb{E}\|\bar{a}_t - \mathbb{E}\bar{a}_t\|^2 \\ &= \|g_t\|^2 + \frac{1}{q}\mathbb{E}\|a_t^{(j)} - g_t\|^2 \\ &= \left(1 + \frac{k+1}{q}\right)\|g_t\|^2.\end{aligned}$$

For the finite-difference remainder, Jensen's inequality gives

$$\mathbb{E}\|\bar{b}_t\|^2 \le \frac{1}{q}\sum_{j=1}^{q}\mathbb{E}\|b_t^{(j)}\|^2 \le \frac{L_F^2\mu^2}{4}M_k.$$

Using $\|\bar{a}_t + \bar{b}_t\|^2 \le 2\|\bar{a}_t\|^2 + 2\|\bar{b}_t\|^2$, we obtain

$$\mathbb{E}\|\widehat{g}_t\|^2 \le 2\left(1 + \frac{k+1}{q}\right)\|g_t\|^2 + \frac{L_F^2\mu^2}{2}M_k.$$

Let $A_k = 1 + (k+1)/q$, then we have

$$\mathbb{E}\|\widehat{g}_t\|^2 \le 2A_k\|g_t\|^2 + \frac{L_F^2\mu^2}{2}M_k.$$

By smoothness of ϕ, the update $\beta_{t+1} = \beta_t - \eta\widehat{g}_t$ satisfies

$$\phi(\beta_{t+1}) \le \phi(\beta_t) - \eta\langle g_t, \widehat{g}_t\rangle + \frac{L_F\eta^2}{2}\|\widehat{g}_t\|^2.$$

Taking conditional expectation and substituting the two bounds above gives

$$\begin{aligned}\mathbb{E}_t[\phi(\beta_{t+1})] \le \phi(\beta_t) &- \eta\left(\frac{1}{2}\|g_t\|^2 - \frac{L_F^2\mu^2}{8}M_k\right) \\ &+ \frac{L_F\eta^2}{2}\left(2A_k\|g_t\|^2 + \frac{L_F^2\mu^2}{2}M_k\right).\end{aligned}$$

Rearranging,

$$\begin{aligned}\mathbb{E}_t[\phi(\beta_{t+1})] \le \phi(\beta_t) &- \left(\frac{\eta}{2} - L_F\eta^2A_k\right)\|g_t\|^2 \\ &+ \frac{\eta L_F^2\mu^2}{8}M_k + \frac{L_F^3\eta^2\mu^2}{4}M_k.\end{aligned}$$

If $0 < \eta \le \frac{1}{8L_FA_k}$, then $\eta/2 - L_F\eta^2A_k \ge \eta/4$. Also, since $A_k \ge 1$, we have $L_F\eta \le 1/8$, and hence

$$\frac{L_F^3\eta^2\mu^2}{4}M_k \le \frac{\eta L_F^2\mu^2}{32}M_k.$$

Therefore,

$$\mathbb{E}_t[\phi(\beta_{t+1})] \leq \phi(\beta_t) - \frac{\eta}{4}\|g_t\|^2 + \frac{5\eta L_F^2 \mu^2}{32} M_k.$$

Taking total expectation and summing from $t = 0$ to $T - 1$, we get

$$\frac{\eta}{4} \sum_{t=0}^{T-1} \mathbb{E}\|g_t\|^2 \leq \phi(\beta_0) - \mathbb{E}[\phi(\beta_T)] + \frac{5\eta L_F^2 \mu^2}{32} M_k T.$$

Since $\phi(\beta_T) \geq \phi_k^{\inf}$, dividing by $\eta T/4$ yields

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla\phi(\beta_t)\|^2 \leq \frac{4(\phi(\beta_0) - \phi_k^{\inf})}{\eta T} + \frac{5}{8} L_F^2 \mu^2 M_k.$$

This proves the theorem. □

A.2 Effect of Subspace Rank

Rank	RTE	BoolQ
1	83.3 ± 3.3	**85.1** ± 0.8
8	**83.9** ± 1.7	84.9 ± 0.6
32	82.2 ± 1.2	84.6 ± 0.7

Table 6: Effect of activation subspace rank on Llama-3-8B. Results are mean ± standard deviation over four seeds $\{0, 13, 42, 100\}$.

We evaluate ranks $r \in 1, 8, 32$ on RTE and BoolQ. Results are averaged over four seeds $0, 13, 42, 100$. As shown in Table 6, performance is relatively insensitive to rank over this range, with no consistent benefit from increasing the subspace dimension. Rank $r = 1$ achieves the highest mean accuracy on BoolQ and remains competitive on RTE, where $r = 8$ improves the mean by only 0.6 points. These results are consistent with the dimension–alignment trade-off discussed in Section 4: increasing the rank may capture additional gradient directions, but also enlarges the coefficient space in which the ZO gradient is estimated.

We therefore use $r = 1$ in the main experiments, as it provides competitive accuracy with the smallest coefficient space and adapter memory footprint.

A.3 Multi-Seed Evaluation

We evaluate three classification tasks over four seeds $\{0, 13, 42, 100\}$, varying both the training subset and the perturbation randomness. ZO-Act achieves the highest mean accuracy on all three tasks (Table 7), exceeding the strong baselines on RTE, CB, and BoolQ, respectively. On GSM8K (Section 5.2), ZO-Act likewise achieves a higher mean accuracy than the strong baselines.

A.4 Evaluation on Qwen Models

We further evaluate ZO-Act on Qwen2.5-7B and Qwen2.5-32B across six tasks, comparing against baseline results reported by Fang et al. (2026). We follow their experimental setup with 1000 training examples, and batch size 16. ZO-Act uses 4000 steps with $q = 8$, totaling 36k forward evaluations, compared with approximately 40k for LOZO+.

As shown in Table 8, ZO-Act achieves the highest average score among the ZO methods at both model sizes. At 7B, it outperforms the ZO baselines on four tasks, performs comparably to LOZO+ on RTE, and scores slightly lower on WSC. It also narrows the average performance gap between ZO methods and full fine-tuning. At 32B, ZO-Act achieves the highest score on all six tasks. These results support the effectiveness of ZO-Act across model families and at a larger model size.

Method	RTE	CB	BoolQ
ZO-Muon	76.62 ± 4.54	71.43 ± 2.92	82.80 ± 1.48
AGZO	75.00 ± 5.53	70.09 ± 1.72	82.85 ± 1.12
LOZO+	77.17 ± 0.95	72.32 ± 2.31	83.25 ± 0.64
ZO-Act	**83.30** ± 3.26	**80.36** ± 3.26	**85.07** ± 0.83

Table 7: Llama-3-8B classification accuracy, reported as mean $\pm$ standard deviation over seeds $\{0, 13, 42, 100\}$.

Model	Method	SST-2	RTE	BoolQ	WSC	WiC	SQuAD	Avg.
Qwen2.5-7B	Zero-shot	63.6	84.5	81.9	68.3	61.0	85.1	74.1
	ICL	95.3	83.4	85.4	66.3	64.4	74.8	78.3
	FT	95.0	89.9	87.6	76.0	75.5	91.5	85.9
	FT-LoRA	79.6	89.5	80.4	73.1	65.8	89.8	79.7
	MeZO	89.1	88.1	84.1	69.2	68.8	87.4	81.1
	MeZO-LoRA	78.9	87.4	82.9	71.2	64.9	81.6	77.8
	SubZero	88.1	88.4	84.1	72.1	69.3	86.9	81.5
	HiZOO	92.0	88.8	82.9	65.4	67.1	87.1	80.6
	LOZO	90.6	86.6	84.3	73.1	70.2	87.5	82.1
	LOZO+	93.1	**89.9**	85.5	**76.9**	70.4	88.4	84.0
	ZO-Act	**94.3**	**89.9**	**85.6**	76.0	**73.2**	**89.2**	**84.7**
Qwen2.5-32B	Zero-shot	61.5	87.0	84.3	80.8	61.8	85.5	76.8
	MeZO	87.5	90.3	86.7	81.7	65.7	87.9	83.3
	MeZO-LoRA	62.0	87.4	84.8	82.7	64.7	84.7	77.7
	SubZero	89.1	90.3	87.7	82.7	68.8	89.6	84.7
	LOZO	89.4	91.0	87.0	84.6	69.3	90.5	85.3
	LOZO+	93.5	91.3	87.9	84.6	71.8	91.6	86.8
	ZO-Act	**94.8**	**91.7**	**88.7**	**85.6**	**74.6**	**91.8**	**87.9**

Table 8: Performance on Qwen2.5-7B and Qwen2.5-32B.

A.5 Gradient Alignment Along the ZO-Act Trajectory

To verify that the gradient-alignment pattern in Figure 1 also holds along the actual ZO-Act optimization path, we repeat the same diagnostic using checkpoints from a ZO-Act run with $r = 1$ on RTE. As shown in Figure 2, activation-informed subspaces consistently capture substantially more gradient energy than weight-SVD and random alternatives throughout training.

B Experiment Setups

In all experiments, ZO-Act is applied to all linear layers except the embedding layers and task-specific linear heads.

Language understanding and question answering. Following the setting of Lang et al. (2026), ZO-Act is evaluated on six standard language understanding and question answering benchmarks: SST-2, RTE, CB, BoolQ, WiC, and SQuAD in SuperGLUE (Wang et al., 2019). F1 score is reported for SQuAD, while accuracy is reported for all other tasks. Experiments are conducted on two full-precision LLMs, Llama-3-8B and OPT-13B, as well as an INT4 quantized Llama-3-8B model. Following prior ZO fine-tuning work (Malladi et al., 2023), we randomly sample 1,000 training examples and 1,000 test examples for each task, and use the same prompts as MeZO (Malladi et al., 2023).

In Table 1, the baseline results for AGZO and LOZO+ are obtained from our own experiments using the same experimental settings as Lang et al. (2026), while all other results are taken from Lang et al. (2026). In Table 8, all baseline results are taken from Fang et al. (2026).

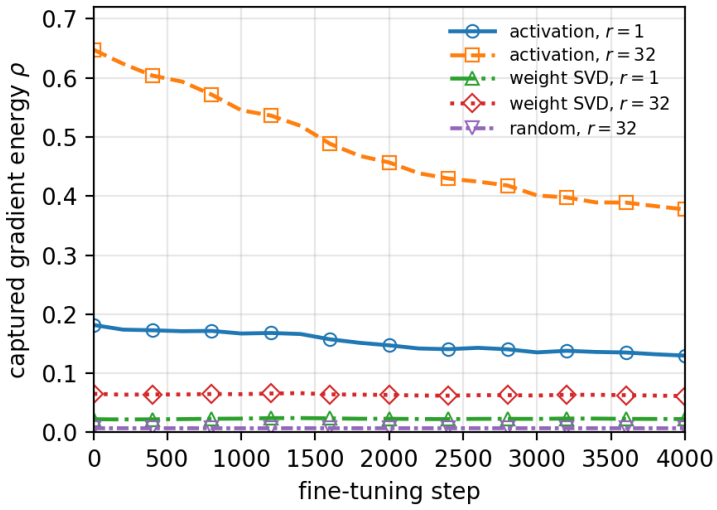


Figure 2: Captured gradient energy along the ZO-Act $r = 1$ training trajectory on RTE with Llama-3-8B.

	CB	RTE	WiC	SST-2	BoolQ	SQuAD	Commonsense	GSM8K
Rank	1							
Queries q	8							
Optimizer	Adam / SGD						Adam	Adam
LR (Adam)	1×10^{-4}	5×10^{-5}	4×10^{-5}	3×10^{-5}	5×10^{-5}	2.5×10^{-5}	2×10^{-5}	1.25×10^{-5}
LR (SGD)	3×10^{-6}	3×10^{-6}	3×10^{-6}	3×10^{-6}	5×10^{-6}	3×10^{-6}	–	–
Training steps	4,000						20,000	10,000
Batch size	16							
Scheduler	constant with warmup							
Weight decay	0							
μ	10^{-3}							

Table 9: Hyperparameters for ZO-Act on Llama-3-8B.

Commonsense reasoning. The commonsense reasoning benchmark consists of eight tasks: ARC-Challenge, ARC-Easy (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SocialIQA (Sap et al., 2019), and WinoGrande (Sakaguchi et al., 2021). ZO-Act is fine-tuned on the combined training data using Llama-3-8B, and performance is reported as accuracy on each individual task as well as the average accuracy across all tasks. ZO-Act and ZO-Muon use 20k training steps with 180k forward passes, and other methods use 100k training steps with 200k forward passes.

Mathematical reasoning. We evaluate ZO-Act on GSM8K (Cobbe et al., 2021), a benchmark of grade-school word problems requiring multi-step arithmetic reasoning. The model generates a complete solution, with accuracy measured by exact match on the final numeric answer. We fine-tune Llama-3-8B on all 7,473 training problems and evaluate on all 1,319 test problems, allowing up to 512 generated tokens per problem. We report the final checkpoint for every method, following the convention for PEFT methods. We train ZO-Act and ZO-Muon for 10k steps and AGZO and LOZO+ for 50k steps, giving ZO-Act a slightly smaller total forward-evaluation budget. We run each configuration with four seeds $\{0, 13, 42, 100\}$.

The hyperparameters are reported in Table 9, Table 10, Table 11, and Table 12.

	CB	RTE	WiC	SST-2	BoolQ	SQuAD	Commonsense
Rank				1			
Queries q				8			
Optimizer				Adam			
LR (Adam)	6×10^{-5}	5×10^{-5}	3.5×10^{-5}	3×10^{-5}	5×10^{-5}	2.5×10^{-5}	2×10^{-5}
Training steps				4,000			20,000
Batch size				16			
Scheduler				constant with warmup			
Weight decay				0			
μ				10^{-3}			

Table 10: Hyperparameters for ZO-Act on INT4 quantized Llama-3-8B.

	CB	RTE	WiC	SST-2	BoolQ	SQuAD
Rank				1		
Queries q				8		
Optimizer				Adam / SGD		
LR (Adam)	3×10^{-5}	2.5×10^{-5}	2×10^{-5}	2.5×10^{-5}	2×10^{-5}	2.5×10^{-5}
LR (SGD)	3×10^{-7}	3×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{-6}
Training steps				4,000		
Batch size				16		
Scheduler				constant with warmup		
Weight decay				0		
μ				10^{-3}		

Table 11: Hyperparameters for ZO-Act on OPT-13B.

	RTE	WiC	SST-2	BoolQ	WSC	SQuAD
Rank				1		
Queries q				8		
Optimizer				Adam		
LR (Qwen2.5-7B)	1×10^{-5}	1×10^{-4}	7×10^{-5}	1×10^{-4}	4.5×10^{-5}	2.5×10^{-5}
LR (Qwen2.5-32B)	3×10^{-5}	1×10^{-4}	1.5×10^{-4}	1×10^{-4}	5×10^{-5}	5×10^{-5}
Training steps				4,000		
Batch size				16		
Scheduler				constant with warmup		
Weight decay				0		
μ				10^{-3}		

Table 12: Hyperparameters for ZO-Act on Qwen2.5 models.