Powering Up Zeroth-Order Training via Subspace Gradient Orthogonalization

Yicheng Lang[1] **Changsheng Wang**[1] **Yihua Zhang**[1] **Mingyi Hong** [2]
Zheng Zhang [3] **Wotao Yin** [4] **Sijia Liu**[1,5]
[1]Michigan State University [2]University of Minnesota [3]University of California, Santa Barbara
[4]DAMO Academy, Alibaba Group US [5]IBM Research, USA

Abstract

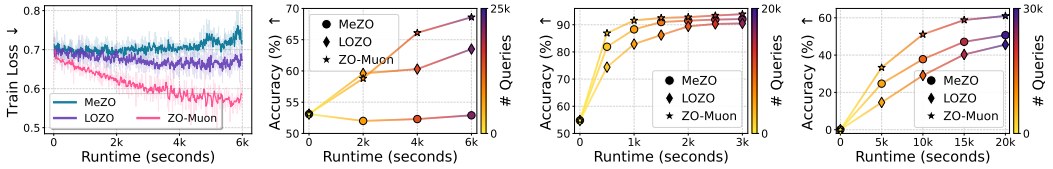
Zeroth-order (ZO) optimization provides a gradient-free alternative to first-order (FO) methods by estimating gradients via finite differences of function evaluations, and has recently emerged as a memory-efficient paradigm for fine-tuning large-scale models by avoiding backpropagation. However, ZO optimization has a fundamental tension between accuracy and query efficiency. In this work, we show that this can be substantially improved by unifying two complementary principles: (i) a projection-based subspace view that reduces gradient estimation variance by exploiting the intrinsic low-rank structure of model updates, and (ii) Muon-style spectral optimization that applies gradient orthogonalization to extract informative spectral structure from noisy ZO gradients. These form a unified framework of subspace gradient orthogonalization, which we instantiate in a new method, **ZO-Muon**, admitting a natural interpretation as a low-rank Muon optimizer in the ZO setting. Extensive experiments on large language models (LLMs) and vision transformers (ViTs) demonstrate that ZO-Muon significantly accelerates convergence and achieves a win–win improvement in accuracy and query/runtime efficiency. Notably, compared to the popular MeZO baseline, ZO-Muon requires only 24.7% of the queries to reach the same SST-2 performance for LLM fine-tuning, and improves accuracy by 25.1% on ViT-L fine-tuning on CIFAR-100. Code is available at `https://github.com/OPTML-Group/ZO-Muon`.

1 Introduction

Training deep models such as LLMs [1–3] relies on backpropagation (BP) to compute first-order (**FO**) gradients for optimization [4, 5]. However, as model sizes scale, the GPU memory cost of BP grows substantially [6]. This motivates a fundamental question with practical impact: Can deep models be trained in a *gradient-free* manner without relying on BP? Toward this end, zeroth-order (**ZO**) optimization [7–9] offers a theoretically grounded alternative to FO optimization by estimating gradients via finite differences of function values using only forward passes of the model, thereby substantially reducing the memory footprint during training [10–13]. Although ZO optimization also arises in black-box or gray-box application settings [14–17] (where FO gradients are infeasible or inaccessible), *memory-efficient fine-tuning of large models* has recently become one of its most impactful and arguably primary applications, as first proposed in [10] and a focus of this work.

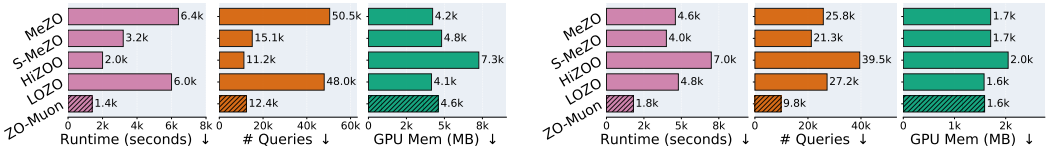
Despite the promise of ZO optimization, achieving convergence and accuracy comparable to FO optimization remains highly nontrivial. A key challenge of ZO optimization is the high variance of query-based gradient estimation, which becomes increasingly severe as the problem dimension grows [8, 9, 17, 20, 21]. Therefore, to reduce the variance of ZO gradient estimation, existing work has focused on designing improved gradient estimators. Examples include preconditioning the estimator with Hessian information [19, 22], optimizing perturbation directions before computing directional derivatives [23], and imposing structural priors such as sparsity [13, 24] or low-rankness [18, 25] in the gradient estimation. However, these designs still primarily operate in the original parameter space of the model, which limits the extent of variance reduction. As a result, a significant performance gap still remains between ZO and FO methods in deep model training.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) *LLM fine-tuning*: OPT-1.3B on RTE (b) *ViT fine-tuning*: ViT-B on CIFAR-10 (left) & 100 (right)

Figure 1: ZO-Muon achieves faster convergence and better fine-tuning accuracy than the SOTA ZO baselines MeZO [10] and LOZO [18] against runtime. *(a)* OPT-1.3B fine-tuned on the RTE task: training loss (*left*) and test accuracy (*right*) vs. runtime, with the cumulative query count indicated by the color bar in the right subplot. *(b)* ViT-B fine-tuned on CIFAR-10 (*left*) and CIFAR-100 (*right*), shown in the same format as in (a, right).



(a) Efficiency comparison for OPT-1.3B fine-tuned on SST-2 (b) Efficiency comparison for ViT-B fine-tuned on CIFAR-10

Figure 2: ZO-Muon yields lower runtime and fewer function queries than all baselines (under comparable GPU memory usage), when reaching the target fine-tuning accuracies: *(a)* 0.90 on SST-2 and *(b)* 0.93 on CIFAR-10. Baselines include MeZO [10], SparseMeZO (S-MeZO) [13], HiZOO [19], and LOZO [18].

Another principled approach to reducing variance in ZO optimization is to use multiple queries (at the cost of increased query complexity) to average out estimation noise. However, a linear increase in the number of queries (or runtime) yields only a *sub-linear* improvement in performance (and may even lead to a multi-query paradox with no improvement) [26], making it challenging to strike a favorable *accuracy-efficiency trade-off*. For example, FZOO [27] requires a substantial number of queries to estimate the variance of function evaluations for gradient normalization. Similarly, SVRG-based ZO methods [21, 28] also require an excessive number of queries to construct error-correction terms for determining descent directions from estimated gradients. ZO-Align [26] leverages projection alignment across multiple queries, but is difficult to scale because it must store multiple ZO gradients in large matrices. MUZO [29] takes not only multiple queries but also the more complex Adam optimizer in ZO optimization, which incurs additional memory overhead.

As mentioned above, although ZO optimization for model training has made substantial recent progress, its core bottlenecks, limited accuracy due to high variance and a poor accuracy-efficiency trade-off, remain unresolved. Thus, we ask:

(Q) *Can we advance ZO training to achieve a win-win in both accuracy and efficiency?*

To address **(Q)**, we propose a novel *subspace gradient orthogonalization* viewpoint to revisit and redesign ZO optimization. This framework consists of two key components: *projection-based subspace ZO optimization* and *gradient-orthogonalized spectral optimization*. The former reduces the variance of ZO gradient estimation by projecting the original full-space problem into a lower-dimensional subspace. The latter further improves optimization accuracy by leveraging gradient orthogonalization in the recent Muon optimizer [30, 31], which can extract meaningful descent directions from noisy ZO gradient estimates rather than directly using their entry-wise values. We term the proposed subspace gradient orthogonalization-based ZO optimization approach "**ZO-Muon**" to echo its connection to Muon. As shown in **Fig. 1** and **Fig. 2**, ZO-Muon consistently outperforms a range of state-of-the-art (SOTA) baselines in both convergence speed and accuracy. Moreover, these performance gains do not incur additional query cost under the same runtime budget, indicating improved efficiency in both runtime and query usage across LLM and ViT fine-tuning. This achieves the desired win-win outcome posed in **(Q)**. We summarize our main **contributions** as follows.

① (Formulation) We revisit ZO optimization from a projection-based perspective and propose a *subspace randomized gradient estimator* (Subspace RGE), which substantially reduces the variance of gradient estimation by operating in a low-dimensional subspace.

② (Algorithm) We show that Muon's *gradient orthogonalization* provides complementary gains when meeting Subspace RGE, yielding an enhanced ZO optimization algorithm, termed **ZO-Muon**.

③ (Practice) Compared with the stateful baselines including MeZO [10], LOZO [18], SparseMeZO [13], HiZOO [19], SubZero [25] and FZOO [27], ZO-Muon consistently delivers superior accuracy and efficiency; see Figs. 1–2 for representative results.

2 Related Work

This work connects three lines of research: (i) ZO optimization and its applications to deep models, (ii) subspace optimization for memory- and compute-efficient training, and (iii) emerging gradient-orthogonalized optimizers for large-scale model training. See detailed review in **Appendix A**. Among existing studies, the most closely related is JAGUAR-Muon [32], which attempts to directly apply gradient orthogonalization (GO) to full-space ZO momentum, but incurs additional memory and runtime overhead over MeZO [10], and shows very limited empirical gains. This is likely because applying GO to noisy, high-dimensional ZO estimates amplifies non-informative spectral components, a limitation also confirmed by our work. By contrast, we show that GO becomes effective for ZO *only when coupled with a subspace formulation*, which we term *subspace gradient orthogonalization*.

3 Rethinking ZO Optimization: A Projected Subspace View

In this section, we first introduce the standard ZO optimization formulation, whose core algorithmic component is a query-based ZO gradient estimator built from finite differences of function values. We then develop a new projection-based subspace perspective, from which we derive a subspace ZO gradient estimator and motivate its effectiveness for improving ZO accuracy and query efficiency.

Standard ZO optimization in the full space. Consider a standard minimization problem:

$$\underset{\mathbf{X}}{\text{minimize}} \quad f(\mathbf{X}), \tag{1}$$

where, instead of vectorizing the variable $\mathbf{X} \in \mathbb{R}^{m\times n}$ (*e.g.*, model weights or weight perturbations to a pre-trained model for fine-tuning), we use its matrix form for notational consistency throughout the paper. Here, f denotes the objective function (*e.g.*, the fine-tuning loss for a downstream task).

To solve problem (1), the standard ZO optimization approach replaces the gradient used in FO methods with a query-based ZO gradient estimator constructed from finite differences of function values [7]. A standard choice is the *randomized gradient estimator* (**RGE**) [9], whose effectiveness for memory-efficient LLM fine-tuning has been demonstrated by MeZO [10], which is built upon ZO-SGD. *A generic form of the RGE* [8] is given by

$$\hat{\nabla}_{\mathbf{X}} f(\mathbf{X}) = \frac{1}{N_{\mathrm{q}}} \sum_{i=1}^{N_{\mathrm{q}}} \frac{f(\mathbf{X}+\mu\mathbf{\Psi}_i) - f(\mathbf{X})}{\mu}\, \mathbf{\Psi}_i, \tag{2}$$

where $\hat{\nabla}_{\mathbf{X}} f(\mathbf{X})$ denotes the estimated gradient of f with respect to (w.r.t.) $\mathbf{X}$, N_{q} denotes the number of queries, $\mathbf{\Psi}_i \in \mathbb{R}^{m\times n}$ is a random perturbation matrix applied to the optimization variable $\mathbf{X}$ whose entries are independently drawn from the standard Gaussian distribution, and $\mu > 0$ is a small smoothing (perturbation) parameter. As shown in (2), the RGE is constructed using the *forward-difference* approximation of function values for each query. An alternative variant adopts the *central-difference* scheme, $\frac{f(\mathbf{X}+\mu\mathbf{\Psi})-f(\mathbf{X}-\mu\mathbf{\Psi})}{2\mu}\,\mathbf{\Psi}$, which is used by MeZO [10]. In this work, we prefer the forward-difference estimator when $N_{\mathrm{q}} > 1$ due to its lower query complexity: the RGE in (2) requires only $(N_{\mathrm{q}}+1)$ function queries, compared to $2N_{\mathrm{q}}$ queries for the central-difference scheme. When $N_{\mathrm{q}} = 1$, the query complexity of the two schemes is the same; therefore, we use the central-difference estimator.

The RGE (2) provides an *unbiased* estimator of the gradient of a *smoothed* version of the original objective, *i.e.*, $\mathbb{E}_{\mathbf{\Psi}}[f(\mathbf{X}+\mu\mathbf{\Psi})]$, and its estimation variance scales as $O(mn/N_{\mathrm{q}})$ [7]. Consequently, a fundamental limitation of RGE is its high gradient estimation variance, which scales proportionally with the problem dimension and can become prohibitively large for training large-scale models. In theory, mitigating this variance requires using a large number of queries N_{q}, leading to an unfavorable variance-query trade-off.

Subspace ZO optimization: A *projection*-based view. A growing body of evidence suggests that, in large model training (especially fine-tuning), the weights and their gradients evolve within a much lower-dimensional subspace [33–36]. Therefore, it is natural to ask *whether one can transition ZO optimization from the full space to a subspace* since the reduced effective dimension can significantly lower the variance of RGE.

Motivated by the above, we propose to approximate the original problem (1) via a projected variant, under the implicit assumption that the optimization variable $\mathbf{X}$ admits a low-dimensional representation after projection. To be concrete, let $\mathbf{P} \in \mathbb{R}^{m\times r}$ denote a projection matrix whose r columns (with $r \ll \min\{m,n\}$) are *orthonormal*, *i.e.*, $\mathbf{P}^\top\mathbf{P} = \mathbf{I}_r$, where $\mathbf{I}_r \in \mathbb{R}^{r\times r}$ denotes the identity matrix.

We then represent the original variable $\mathbf{X}$ using a lower-dimensional representation $\mathbf{Z} \in \mathbb{R}^{r\times n}$ over the subspace spanned by the columns of $\mathbf{P}$, yielding the approximation $\mathbf{X} \approx \mathbf{PZ}$. We choose $\mathbf{P}$ to be a *random column-orthonormal* projection matrix (that can be readily obtained via QR decomposition of a random Gaussian matrix) and approximate (1) as:

$$\underset{\mathbf{Z}\in\mathbb{R}^{r\times n}}{\text{minimize}} \ \mathbb{E}_{\mathbf{P}}[f(\mathbf{PZ})]\,. \tag{3}$$

The above provides a $\mathbf{P}$-projected view of (1) that can be used to guide effective ZO optimization.

To further understand the relationship between the problem (1) and its projected approximation (3), we examine the FO (stochastic) gradient of the latter w.r.t. the lower-dimensional variable $\mathbf{Z}$:

$$\underbrace{\nabla_{\mathbf{Z}} f(\mathbf{PZ}) = \mathbf{P}^\top \nabla_{\mathbf{X}} f(\mathbf{X})}_{\text{① gradient in } \mathbf{Z} \text{ space}},\ \underbrace{\mathbf{P}\nabla_{\mathbf{Z}} f(\mathbf{PZ}) = \mathbf{P}\mathbf{P}^\top \nabla_{\mathbf{X}} f(\mathbf{X})}_{\text{② gradient lifting back to } \mathbf{X} \text{ space}}. \tag{4}$$

Here, the transition from ① to ② indicates that the subspace gradient $\nabla_{\mathbf{Z}} f(\mathbf{PZ})$, when lifted back to the full space via the same projection matrix $\mathbf{P}$, yields a projected full-space gradient $\mathbf{P}\mathbf{P}^\top \nabla_{\mathbf{X}} f(\mathbf{X})$. Defining $\hat{\mathbf{P}} := \mathbf{P}\mathbf{P}^\top \in \mathbb{R}^{m\times m}$, we note that $\hat{\mathbf{P}}$ is a projection matrix that maps any vector onto the column space of $\mathbf{P}$ and satisfies the idempotence property $\hat{\mathbf{P}}^2 = \hat{\mathbf{P}}$ (due to $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$).

The key implications of (4) for ZO optimization are as follows. Under the projected formulation (3), we can apply the RGE in (2) directly in the reduced $\mathbf{Z}$-space to obtain a dimension-reduced (and hence variance-reduced) ZO gradient estimate $\hat{\nabla}_{\mathbf{Z}} f(\mathbf{PZ})$, corresponding to the ZO realization of step ① in (4). This is given by

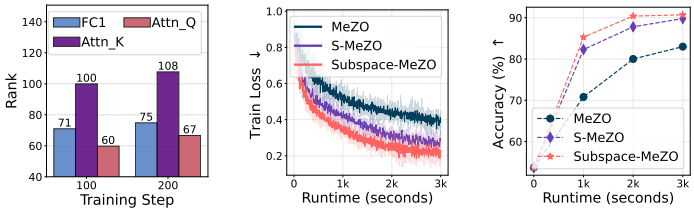
$$\hat{\nabla}_{\mathbf{Z}} f(\mathbf{PZ}) = \frac{1}{N_{\mathrm{q}}} \sum_{i=1}^{N_{\mathrm{q}}} \frac{f(\mathbf{X} + \mu\mathbf{P}\mathbf{\Psi}_i) - f(\mathbf{X})}{\mu}\, \mathbf{\Psi}_i, \tag{5}$$

where in contrast to the standard RGE in (2), the perturbation $\mathbf{\Psi}_i \in \mathbb{R}^{r\times n}$ is applied to the lower-dimensional subspace variable $\mathbf{Z}$, and we write the function value $f(\mathbf{PZ})$ using the exact correspondence between $\mathbf{PZ}$ and $\mathbf{X}$.

The advantage of the ZO gradient estimator in (5) w.r.t. $\mathbf{Z}$ lies in its reduced estimation variance. To obtain a full-space gradient estimate, we next leverage step ② in (4) to lift the subspace ZO gradient in (5) back to the original space, yielding a ZO gradient estimate w.r.t. $\mathbf{X}$, which we term **Subspace RGE** due to its reliance on (5):

$$\hat{\nabla}_{\mathbf{X}} f(\mathbf{X}) \approx \mathbf{P}\hat{\nabla}_{\mathbf{Z}} f(\mathbf{PZ}) = \mathbf{P}\left[\frac{1}{N_{\mathrm{q}}} \sum_{i=1}^{N_{\mathrm{q}}} \frac{f(\mathbf{X} + \mu\mathbf{P}\mathbf{\Psi}_i) - f(\mathbf{X})}{\mu}\, \mathbf{\Psi}_i\right] = \frac{1}{N_{\mathrm{q}}} \sum_{i=1}^{N_{\mathrm{q}}} \frac{f(\mathbf{X} + \mu\mathbf{P}\mathbf{\Psi}_i) - f(\mathbf{X})}{\mu}\, \mathbf{P}\mathbf{\Psi}_i. \tag{6}$$

As indicated by (4), the Subspace RGE (6) serves as the ZO counterpart of the projected FO gradient $\mathbf{P}\mathbf{P}^\top \nabla_{\mathbf{X}} f(\mathbf{X})$, rather than of the full gradient. This seems to introduce an estimation bias at first glance, yet in practice it can actually lead to improved performance due to two key factors: (i) the variance reduction benefit enjoyed by (5), and (ii) the intrinsic subspace structure of gradients during model training which makes subspace projection effectively lossless, as will be validated below.



(a) Rank at FO training (b) Training loss (c) Fine-tune accuracy

Figure 3: The advantages of Subspace RGE, illustrated by fine-tuning OPT-1.3B on SST-2: (a) The low-rank structure of model components during FO training with Adam across steps; (b) Performance of Subspace RGE–based MeZO (*i.e.*, Subspace-MeZO) vs. the baselines SparseMeZO (*i.e.*, S-MeZO) and MeZO; And (c) accuracy vs. runtime.

Fig. 3 provides a motivating example based on fine-tuning OPT-1.3B on the SST-2 task, which illustrates the above rationale for using the Subspace RGE in (6) instead of the vanilla RGE in (2). Fig. 3(a) shows that the FO gradients during Adam-based training exhibit a pronounced low-rank structure, where the effective rank is measured by the number of top singular values required to capture 99.99% of the spectral energy (see **Appendix B** for details). In addition, we refer to the integration of Subspace RGE (at $N_q = 1$) into MeZO as **Subspace-MeZO**, and include MeZO [10] and SparseMeZO [13] as baselines; see **Sec. 5.1** for experimental setups. As shown, Subspace-MeZO achieves both faster convergence (Fig. 3(b)) and better accuracy (Fig. 3(c)).

Comparison with prior work: A closer look at Subspace RGE. The last equality in (6) suggests that Subspace RGE can be implemented in the same form as the standard RGE (2), but using $\mathbf{P}\boldsymbol{\Psi}$ as the effective perturbation which admits a low-rank factorization into $\mathbf{P} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{\Psi} \in \mathbb{R}^{r\times n}$ since $r \ll \min\{m, n\}$. Here, we omit the query index of $\boldsymbol{\Psi}$ for brevity. This perspective connects the proposed Subspace RGE to the existing low-rank ZO optimization approach, **LOZO** [18]. We provide a detailed comparison between our proposal and LOZO below.

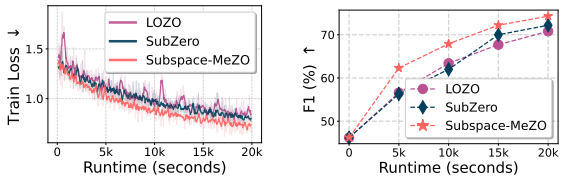
LOZO [18] incorporates low-rank perturbations into the (central-difference) RGE in (2), yielding the *low-rank gradient estimator* (LGE).

$$\hat{\nabla}_{\mathbf{X}} f(\mathbf{X}) = \frac{f(\mathbf{X} + \mu\mathbf{AB}) - f(\mathbf{X} - \mu\mathbf{AB})}{2\mu}\mathbf{AB}, \tag{7}$$

where $\mathbf{A} \in \mathbb{R}^{m\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times n}$ are two random matrices from standard Gaussian distributions with $r \ll \min\{m, n\}$. The overall perturbation $\mathbf{AB}$ thus gives a low-rank, factorized approximation of the full-space perturbation $\boldsymbol{\Psi}$ in (2).

A comparison between the LGE in (7) and the proposed Subspace RGE in (6) shows that the two estimators take essentially the same form, up to a swap between the left perturbation matrices $\mathbf{A}$ and $\mathbf{P}$, and between the right perturbation matrices $\mathbf{B}$ and $\boldsymbol{\Psi}$. However, beyond this formal similarity, there exists a *key difference* in the choice of the (left) perturbation matrix: Subspace RGE enforces $\mathbf{P}$ to be *column-orthonormal*, whereas LGE imposes no such constraint on $\mathbf{A}$. This distinction is critical because $\mathbf{P}$ in Subspace RGE admits a natural interpretation as a subspace projector, which is precisely why it estimates the projected gradient $\hat{\mathbf{P}}\nabla_{\mathbf{X}} f(\mathbf{X})$ in step ② of (4), where $\hat{\mathbf{P}} = \mathbf{P}\mathbf{P}^\top$. Recall that $\hat{\mathbf{P}}$ is a projection matrix satisfying the idempotence property, and it is easy to verify that $(\mathbf{I} - \hat{\mathbf{P}})$ is also a projection matrix onto the subspace orthogonal to the column space of $\mathbf{P}$. The above indicates that the full gradient naturally admits the decomposition $\nabla_{\mathbf{X}} f(\mathbf{X}) = \hat{\mathbf{P}}\nabla_{\mathbf{X}} f(\mathbf{X}) + (\mathbf{I} - \hat{\mathbf{P}})\nabla_{\mathbf{X}} f(\mathbf{X})$, where $(\mathbf{I} - \hat{\mathbf{P}})$ is the complementary projection. Therefore, the use of a column-orthonormal matrix $\mathbf{P}$ in Subspace RGE endows the ZO method with a clear geometric interpretation about the FO gradient component it aims to estimate. By contrast, LOZO does not admit such an interpretation.

In addition to LOZO, SubZero [25] also develops a low-rank perturbation-based RGE similar to (7). Although, like us, SubZero considers column-orthonormal projections motivated by subspace optimization [37, 38], its use of projection is conceptually different from our viewpoint. In our framework, as formalized by steps ①–② and the derivation of (6), we treat subspace optimization as an explicit intermediate step: we first perform RGE in a reduced subspace and then lift the resulting gradient estimate back to the full space to solve the original ZO problem. Furthermore, SubZero uses two projection matrices, whose QR-based construction introduces nontrivial runtime overhead and may degrade accuracy due to double projection. In **Fig. 4**, we validate the advantage of the proposed subspace RGE within the Subspace-MeZO framework when fine-tuning OPT-13B on the SQuAD task. As shown, even when compared against LOZO and SubZero, Subspace-MeZO achieves the fastest convergence (Fig. 4(a)) and the best fine-tune performance (Fig. 4(b)) with the same runtime.



(a) Training loss vs. runtime (b) Fine-tune performance

Figure 4: Training (a) and testing (b) performance comparison of the proposed Subspace RGE–based MeZO (*i.e.*, Subspace-MeZO) with existing low-rank perturbation–based ZO baselines, LOZO [18] and SubZero [25], for fine-tuning OPT-13B on SQuAD whose performance is measured by F1.

4 ZO-Muon: Boosting Subspace ZO via Gradient Orthogonalization

In this section, we show that *gradient orthogonalization* (**GO**), the core ingredient underlying the recently popular optimizer Muon [30, 31], can be used to further boost the subspace ZO method.

Gradient orthogonalization in Muon: From FO to ZO, with insights from the failure of ZO-Muon-V0. Before introducing ZO-Muon, we briefly recall the key idea behind FO Muon. Muon can be interpreted as a steepest-descent method under a spectral-norm constraint [39], or more broadly as a spectral optimization method that exploits the *matrix-level spectral structure* of descent directions rather than their *entry-wise* information. Concretely, given the iterate $\mathbf{X}_t$ of the optimization variable $\mathbf{X}$ at iteration t and the learning rate η_t, the **basic Muon** (without momentum) takes the form [30]:

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \eta_t \,\mathrm{msign}(\mathbf{G}_t), \tag{8}$$

where $\mathbf{G}_t$ denotes the (stochastic) gradient at iteration t, and $\mathrm{msign}(\cdot)$ denotes the matrix sign function, also known as matrix whitening transformation. The quantity $\mathrm{msign}(\mathbf{G}_t)$ can be interpreted as the descent direction obtained by projecting $\mathbf{G}_t$ onto the spectral-norm unit ball [40], which admits the closed-form expression:

$$\mathrm{msign}(\mathbf{G}) = \mathbf{G}(\mathbf{G}^\top \mathbf{G})^{-\frac{1}{2}} = \mathbf{U}\,\mathrm{sign}(\mathbf{\Sigma})\,\mathbf{V}^\top = \mathbf{U}_{[:,:k]}\,\mathbf{V}_{[:,:k]}^\top, \tag{9}$$

where $\mathbf{G} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the singular value decomposition (SVD) of $\mathbf{G}$, and $\mathrm{sign}(\cdot)$ denotes the (entry-wise) sign function, returning 1 for non-zero (positive) singular values and 0 otherwise. Moreover, $\mathbf{U}_{[:,:k]} \in \mathbb{R}^{m\times k}$ and $\mathbf{V}_{[:,:k]} \in \mathbb{R}^{n\times k}$ denote the submatrices of $\mathbf{U}$ and $\mathbf{V}$ consisting of their first k columns, respectively, where k is the *rank* of $\mathbf{G}$.

As shown in (9), the msign operation plays the role of GO by equalizing all active singular directions and producing a spectrally isotropic update. This leads to the key benefit in Muon, which enables the model update to fully explore and exploit the spectral descent directions. In addition, Muon provides an efficient numerical approximation of $\mathrm{msign}(\cdot)$ via Newton–Schulz (**NS**) iterations [39], avoiding the need to compute a full SVD.

Inspired by Muon's spectral advantage, one may naturally ask whether exploiting such spectral information in the ZO gradient estimator can also bring similar benefits as in FO optimization. Accordingly, by replacing $\mathbf{G}_t$ with its standard RGE counterpart in (2), we obtain a vanilla ZO-Muon variant, which we denote as **ZO-Muon-V0**, *i.e.*, applying GO directly to MeZO.

However, we find that ZO-Muon-V0 does *not* provide additional benefits over the GO-*absent* baseline, MeZO. As shown in **Fig. 5**, when fine-tuning OPT models on SST-2, ZO-Muon-V0 *underperforms* MeZO in both accuracy and runtime efficiency under the same query budget. This *accuracy* underperformance is likely because directly applying GO to noisy ZO gradient estimates amplifies spurious directions during matrix whitening, thereby degrading optimization performance. From a *runtime* perspective, performing the NS iterations for GO also introduces additional computational overhead compared to MeZO, *i.e.*, RGE-based ZO-SGD. In addition to ZO-Muon-V0, we also evaluate the ineffectiveness of JAGUAR-Muon [32], another direct attempt to integrate Muon into ZO optimization; see **Fig. A1** in **Appendix C**.

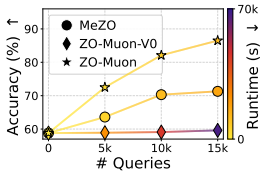


Figure 5: Demonstration of the ineffectiveness of ZO-Muon-V0 vs. MeZO and the effectiveness of ZO-Muon through fine-tuning OPT-13B on SST-2.

ZO-Muon: How and why. Effectively harnessing Muon's spectral advantages in the ZO setting is *nontrivial*. We show that the subspace ZO formulation (Sec. 3) provides the essential structure to realize these benefits; This is *not a simple integration*, but a necessary design choice. Specifically, subspace RGE (5) produces higher-quality, variance-reduced gradient estimates before GO. This is critical, as GO induces spectrally isotropic updates that can otherwise amplify noise in low-quality ZO estimates, leading to degraded performance (Fig. 5). Moreover, since the cost of the msign operation in (9) scales with matrix rank [41], the (low-rank) subspace structure also improves efficiency. As validated by ZO-Muon in Fig. 5, this synergy yields a clear accuracy–efficiency win–win.

We next formally develop ZO-Muon. Recalling the projection-based view used to derive Subspace RGE in (6) from (4), we can integrate GO (*i.e.*, the msign operation) with the subspace RGE in (5) on the lower-dimensional variable $\mathbf{Z}$, and then lift the resulting gradient-orthogonalized RGE back to the full space w.r.t. the original variable $\mathbf{X}$. This leads to **ZO-Muon** that modifies the Muon (8) to

$$\mathbf{O}_t = \overbrace{\mathbf{P}\mathrm{msign}\left(\underbrace{\frac{1}{N_\mathrm{q}}\sum_{i=1}^{N_\mathrm{q}} \frac{f(\mathbf{X}_t + \mu\mathbf{P}\mathbf{\Psi}_i) - f(\mathbf{X}_t)}{\mu}\,\mathbf{\Psi}_i}_{\text{RGE } \hat{\nabla}_\mathbf{Z} f \text{ within subspace in (5), like step ① in (4)}}\right)}^{\text{Lifting GO update back to } \mathbf{X} \text{ space, like step ② in (4)}}, \quad \mathbf{X}_{t+1} = \mathbf{X}_t - \eta_t\,\mathbf{O}_t. \tag{10}$$

Here, compared to (8), $\mathbf{O}_t$ can be understood as the projection-based realization of the gradient-orthogonalized subspace RGE. See a full algorithmic description of ZO-Muon in **Appendix D**.

Furthermore, we provide a theoretical underpinning of the proposed ZO-Muon (10) from a low-rank Muon perspective. As shown, the RGE inside the msign operator prior to GO corresponds to an approximation of the subspace-projected gradient $\mathbf{P}^\top\nabla_\mathbf{X} f(\mathbf{X})$, as given by (5) and the step ① in (4).

Therefore, the FO interpretation of the ZO-Muon step yields

$$\mathbf{O} = \mathbf{P}\mathrm{msign}(\mathbf{P}^\top \mathbf{G}), \quad \mathbf{G} = \nabla_{\mathbf{X}} f(\mathbf{X}), \tag{11}$$

where, for notational simplicity, we omit the iteration index t. Clearly, in the absence of the projection $\mathbf{P} \in \mathbb{R}^{m\times r}$, the step (11) reduces to the standard GO update in Muon (8). Since $r \ll \min\{m, n\}$, the GO step (11) is applied to the low-rank matrix $\mathbf{P}^\top \mathbf{G}$, rather than to the full gradient $\mathbf{G} \in \mathbb{R}^{m\times n}$. Proposition 1 below shows *when* the GO step (11) becomes equivalent to to the original GO step (9).

Proposition 1 *If the projection is chosen as* $\mathbf{P} = \mathbf{U}_{[:,:k]} \in \mathbb{R}^{m\times k}$*, obtained from the SVD of* $\mathbf{G}$ *in (9), then the projection for gradient orthogonalization is* lossless*. That is,*

$$\mathbf{P}\mathrm{msign}(\mathbf{P}^\top \mathbf{G}) = \mathrm{msign}(\mathbf{G}). \tag{12}$$

Proof: See details in **Appendix E**. □

Prop. 1 implies that ZO-Muon can be viewed as a ZO approximation of low-rank Muon (11) [41]. Under standard assumptions for non-convex optimization, **Appendix F** shows that, by treating the ZO gradient estimate as an unbiased estimator of the smoothed objective [7] and interpreting ZO-Muon as its low-rank solver, convergence to stationarity is guaranteed with a sublinear rate of $\mathcal{O}\left(\frac{1}{\delta}\sqrt{\frac{r}{T}}\right)$, where $\delta < 1$ is the subspace alignment ratio measuring the proportion of the projected gradient relative to the full gradient as defined in (4), T the number of iterations, and r the projection dimension. This highlights a key trade-off: projection incurs a factor $1/\delta$ due to partial gradient alignment, but significantly reduces estimation variance, yielding improved dependence on the intrinsic dimension r rather than the original variable dimension.

Projection matrix sampling strategy: Why random is used. Based on Prop. 1, the projection matrix ideally comes from the SVD of FO gradients. In ZO settings, however, this is infeasible. Motivated by the empirical success of using a random column-orthonormal projection matrix $\mathbf{P}$ in Subspace RGE (6), we adopt a practical strategy of lazily sampling $\mathbf{P}$ as a random column-orthonormal projection. While one might think that an informed projection matrix $\mathbf{P}$ in (10), such as one derived from the moving average of past ZO gradients, would better align with the true gradient's principal components, our empirical results suggest otherwise. As detailed in **Table A3 of Appendix I**, Gaussian sketching [42] to past descending directions yields no notable performance gains over random sampling with either Subspace-MeZO or ZO-Muon. Furthermore, maintaining the historical momentum matrix increases GPU memory consumption, which counteracts the primary memory-efficiency motivation of ZO fine-tuning.

Hyperparameters in ZO-Muon: Query number N_q and projection rank r. Two hyperparameters play a central role in ZO-Muon: the number of queries N_q in (10) and the projection rank r of $\mathbf{P}$.

First, we find that setting the query number $N_q > 1$ is crucial for ZO-Muon, *e.g.*, $N_q = 4$ works well in our experiments, indicating that the RGE in (10) should be implemented in its multi-query form. Recall that in the absence of msign, the method reduces to Subspace RGE, yielding Subspace-MeZO as examined in Fig. 3. For both Subspace-MeZO and its full-space baseline MeZO [10], the default choice is $N_q = 1$. This is due to the *multi-query paradox* [26]: Although using multiple queries per iteration can reduce the variance of the gradient estimate, whether it improves optimization performance depends on how the queries are aggregated. In particular, simple averaging with $N_q > 1$ can lead to diminishing returns (*i.e.*, the multi-query paradox) compared to using a single query.

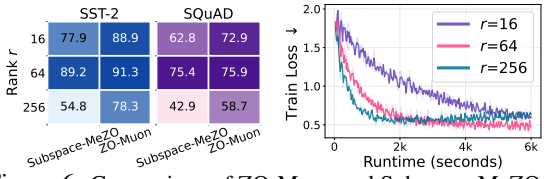


Figure 6: Comparison of ZO-Muon and Subspace-MeZO for fine-tuning OPT-1.3B on SST-2 and SQuAD with $r \in \{16, 64, 256\}$. *(Left)* Accuracy (shown by cell color and numeric value) for different r. ZO-Muon consistently outperforms Subspace-MeZO, with $r = 64$ achieving the best performance. *(Right)* Training loss of ZO-Muon under different r for fine-tuning on SQuAD.

However, unlike Subspace-MeZO, when $N_q = 1$ is used in ZO-Muon, the GO step in (10) reduces to $\mathrm{msign}(s\mathbf{\Psi}) = \mathrm{sign}(s)\mathrm{msign}(\mathbf{\Psi})$, where $s = \frac{f(\mathbf{X}+\mu\mathbf{P}\mathbf{\Psi})-f(\mathbf{X})}{\mu}$ is the finite-difference scalar and $s\mathbf{\Psi}$ is the RGE. Here, $\mathrm{sign}(s)$ returns 1 or -1 depending on the sign of s. In this case, even when the RGE magnitude s becomes small (towards convergence), the msign operation discards its scale and may amplify small noise by enforcing a spectrally isotropic update. Therefore, ZO-Muon benefits from using $N_q > 1$. Meanwhile, it effectively overcomes the multi-query paradox compared to Subspace-MeZO (see **Appendix G** for empirical validation).

For the other key hyperparameter, the projection rank r that defines the subspace via $\mathbf{P}$, we find that ZO-Muon remains effective as long as it is set to a reasonably reduced dimension (*e.g.*, $r \in \{64, 128\}$). This is roughly aligned with the effective rank of FO gradients shown in Fig. 3(a). To validate this, **Fig. 6** compares the performance of ZO-Muon with Subspace-MeZO under different values of r when fine-tuning OPT-1.3B on SST-2 and SQuAD. As shown in **Fig. 6(a)**, ZO-Muon consistently outperforms Subspace-MeZO across all rank choices (indicating the usefulness of GO in ZO-Muon) and achieves the best performance with a moderate rank such as $r = 64$. Setting r too small or too large degrades the performance of ZO-Muon: A too-small r discards informative gradient components during projection, while an overly large r amplifies noisy directions after GO. This is further supported by **Fig. 6(b)**, which shows that ZO-Muon converges slowly when r is too small, whereas using an overly large rank (*e.g.*, $r = 256$) leads to instability and loss increase in the later stage of training, likely due to excessive gradient noise in ZO optimization.

5 Experiments

5.1 Experiment setups

Models and datasets. We evaluate the performance of ZO on fine-tuning tasks for both LLMs and ViTs. For LLMs, we primarily consider LlaMA3-8B [43] and OPT-1.3B/13B [2] on the SuperGLUE benchmark [44]. For ViTs, we fine-tune ViT-B/16 and ViT-L/16 [45] on CIFAR-10/100 [46].

Baselines and evaluations. We compare the proposed ZO-Muon with the following stateful ZO optimization baselines, which were primarily developed for large-model fine-tuning. These include MeZO [10], Sparse-MeZO (S-MeZO) [13], HiZOO [19], LOZO [18], SubZero [25], FZOO [27], and Subspace-MeZO developed in Sec. 3 prior to integration with Muon. We also include FO baselines, full parameter fine-tuning (Full), LoRA [34], and Galore [35]. Following previous works [10, 45], we use Adam [5] for LLMs and SGD (with momentum) [47] for ViTs.

Table 1: ZO-Muon outperforms the ZO baselines by a non-negligible margin, demonstrated by fine-tuning LLMs (LlaMA3-8B and OPT-13B) on SuperGLUE benchmark and fine-tuning ViTs (ViT-B and ViT-L) on CIFAR datasets. FO methods (marked by •) include Adam and LoRA. The best ZO fine-tuning result for each task is highlighted in **bold**.

Method	LLM Fine-tuning						ViT Fine-tuning	
	SST-2	RTE	CB	BoolQ	WiC	SQuAD	CF10	CF100
	LlaMA3-8B						ViT-B	
• Full	96.0	92.0	92.0	86.6	72.6	90.4	98.9	91.2
• LoRA	95.0	80.9	73.2	86.4	70.7	89.4	98.3	86.4
• Galore	95.4	85.2	87.5	85.1	72.7	89.5	97.7	87.3
MeZO	92.7	74.4	**69.6**	76.7	57.8	86.7	93.5	64.5
S-MeZO	92.1	69.7	69.6	80.5	56.9	87.5	93.7	64.3
HiZOO	93.5	75.1	69.6	80.0	59.7	87.3	93.2	64.9
LOZO	92.5	66.8	**69.6**	79.4	55.8	**89.0**	93.2	61.8
SubZero	92.1	71.4	67.9	82.0	58.8	88.3	41.1*	7.3*
Subspace-MeZO	92.3	68.6	**69.6**	80.0	62.9	84.5	94.2	58.2
FZOO	**94.3**	77.6	**69.6**	81.8	60.8	87.9	8.5*	1.7*
ZO-Muon	**94.3**	**81.2**	**69.6**	**82.9**	**65.2**	88.2	**94.9**	**72.6**
	OPT-13B						ViT-L	
• Full	95.3	80.9	94.6	83.5	66.3	89.5	98.8	91.7
• LoRA	94.8	78.3	69.6	80.2	64.3	88.0	98.8	90.6
• Galore	95.4	78.7	71.4	79.5	63.9	88.0	97.8	90.3
MeZO	91.4	66.1	66.0	66.1	59.4	81.8	95.6	47.3
S-MeZO	90.4	63.5	69.6	66.4	58.8	80.8	96.9	56.3
HiZOO	92.1	69.3	69.6	67.6	59.4	82.1	96.3	46.6
LOZO	91.7	70.4	69.6	71.9	60.2	**84.9**	95.3	48.9
SubZero	92.1	71.8	**71.4**	70.8	60.8	84.5	41.2*	3.2*
Subspace-MeZO	91.7	70.7	**71.4**	68.1	**61.7**	83.5	97.0	64.7
FZOO	**93.7**	71.7	69.6	72.2	60.5	84.8	13.0*	1.1*
ZO-Muon	92.5	**73.3**	**71.4**	**72.4**	**61.7**	84.5	**97.2**	**72.4**

* SubZero and FZOO, originally used for LLM tuning, underperform on vision tasks even after extensive hyperparameter tuning.

Implementation details. For ZO-Muon, we perform a hyperparameter grid search over the projection rank $r \in \{64, 128\}$ across all settings, following the rank selection rationale in Sec. 4. We also search over $N_q \in \{4, 8, 16\}$, consistent with our justification for using $N_q > 1$ in Sec. 4. Empirically, we find that setting $N_q = 16$ and $r = 64$ for ViTs, and using $N_q = 4$ with $r \in \{64, 128\}$ for LLMs depending on the task, consistently yields strong performance. To ensure a fair comparison with other baselines, we scale the number of training steps inversely with N_q, so that ZO-Muon does not incur additional query or runtime overhead. The projection matrix $\mathbf{P}$ is resampled via QR decomposition every 100 iterations (*i.e.*, lazily updated). For the msign operation, we use the NS (Newton–Schulz) iteration unless otherwise specified. We set the training batch size to 16 for all LLM experiments, and to 64 and 256 for ViT fine-tuning on CIFAR-10 and CIFAR-100, respectively. The rationale for these choices is supported by the experimental results presented later. Additional experiment setups, including settings of baselines, are detailed in Appendix H.

5.2 Experiment results

ZO-Muon achieves superior ZO fine-tuning accuracy for both LLMs and ViTs. In **Tab. 1**, we compare the LLM and ViT fine-tuning performance of ZO-Muon with FO and ZO baselines

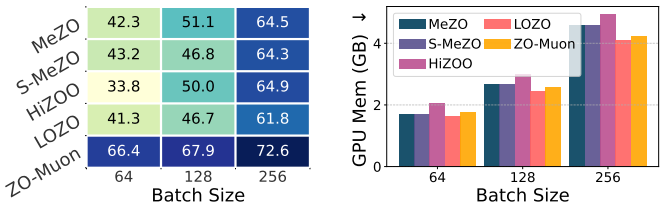
on LlaMA3-8B and OPT-13B, as well as ViT-B and ViT-L, where each column corresponds to a fine-tuning task. As shown, ZO-Muon consistently improves fine-tuning accuracy over existing ZO baselines. In particular, for higher-complexity tasks such as RTE and WiC in LLM fine-tuning and CIFAR-100 in ViT fine-tuning, ZO-Muon achieves substantial gains, narrowing the gap to FO baselines. Notably, on ViT-L CIFAR-100 fine-tuning, ZO-Muon outperforms the second-best ZO method (Subspace-MeZO) by 7.7% and MeZO by 25.1% in accuracy. In addition, we observe that Subspace-MeZO consistently ranks among the top-performing ZO baselines, and that ZO-Muon further improves upon it. This indicates that subspace ZO and Muon's gradient orthogonalization provide complementary benefits for improving overall ZO optimization.

ZO-Muon delivers a strong efficiency profile over other ZO baselines. In **Tab. 2**, we compare the efficiency of ZO-Muon with ZO and FO baselines in terms of the number of optimization steps, number of queries, runtime, memory cost, and GPU usage for fine-tuning LlaMA3-8B on SST-2, with the corresponding accuracy results reported in Tab. 1. For a fair comparison, we fix the total query budget for gradient estimation during fine-tuning to 40k for all ZO methods. As we can see, ZO-Muon significantly reduces the number of ZO optimization steps and thereby yields the improved runtime efficiency compared to other ZO baselines. For example, ZO-Muon requires only 81.1% of MeZO's runtime while outperforming it by 1.6% in fine-tuning accuracy. In addition, compared to FO approaches, ZO optimization does not offer a runtime advantage, but it significantly reduces memory cost and GPU requirements. This is the primary motivation for using ZO fine-tuning as a memory-efficient alternative to FO. We observe that ZO-Muon inherits this advantage and does not increase memory cost or GPU usage compared to other ZO baselines. It is worth noting that while Galore requires less memory than LoRA in theory [35], it consumes higher GPU peak memory in large model fine-tuning, likely due to the SVD operations inovlved in Galore.

Table 2: Efficiency comparison in terms of the number of train steps, query number, runtime, memory cost, GPU usage, and accuracy for fine-tuning LlaMA3-8B on SST-2. Note that the total query budget is fixed to 40k for all ZO methods.

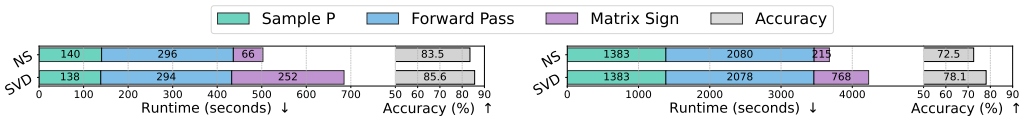
Method	Steps ↓	Queries ↓	Runtime ↓	Memory ↓	GPUs ↓	Accuracy ↑
• Full	625	625	22min	4 × 32.8 GB	4 × A6000	96.0
• LoRA	625	625	14min	2 × 27.9 GB	2 × A6000	95.0
• Galore	625	625	46min	4 × 21.8 GB	4 × A6000	95.4
MeZO	20k	40k	4h 52min	21.0GB	1 × A6000	92.7
S-MeZO	20k	40k	9h 8min	21.3GB	1 × A6000	92.1
HiZOO	13.3k	40k	7h 10min	39.3GB	1 × A6000	93.5
LOZO	20k	40k	5h 4min	20.0GB	1 × A6000	92.5
SubZero	20k	40k	5h 34min	20.2GB	1 × A6000	92.1
Subspace-MeZO	20k	40k	4h 20min	19.2GB	1 × A6000	92.3
FZOO	8k	40k	5h 20min	20.1GB	1 × A6000	94.3
ZO-Muon	8k	40k	3h 57min	19.4 GB	1 × A6000	94.3

ZO-Muon does *not* rely on large batch sizes. In ZO optimization, increasing the batch size reduces gradient-estimation variance but also raises memory usage, since GPU cost is dominated by forward passes. We therefore study ZO performance as a function of batch size. **Fig. 7** compares accuracy and GPU memory of ZO-Muon and baselines for ViT-B fine-tuning on CIFAR-100 under batch sizes 64, 128, and 256. While larger batches generally improve ZO accuracy, reducing the batch size from 256 to 64 causes a severe drop for baselines (*e.g.*, 22.2% for MeZO) but only a modest drop for ZO-Muon (6.2%). Notably, ZO-Muon with batch size 64 still outperforms MeZO with batch size 256. Together with Fig. 7(b), which shows the higher memory cost of larger batches, ZO-Muon is substantially more memory-efficient. The same advantage is also observed for LLM fine-tuning in **Fig. A6**.



(a) Fine-tune accuracy (b) GPU memory

Figure 7: Accuracy and GPU memory comparison under varying batch sizes. ZO-Muon vs. baselines on ViT-B fine-tuning for CIFAR-100 with a fixed query budget.



(a) OPT-*1.3B* fine-tuning (b) OPT-*13B* fine-tuning

Figure 8: Runtime and accuracy comparison of ZO-Muon using NS and SVD for fine-tuning OPT-1.3B/13B on SST-2 under the same query budget (5,000).

SVD vs. Newton–Schulz in ZO-Muon. Thanks to the subspace projection in ZO-Muon (10), the msign operation, previously the main computational bottleneck in Muon, can now be computed via exact SVD as in (9), instead of using NS iterations [30]. Thus, **Fig. 8** compares the runtime and accuracy of ZO-Muon using NS (the default choice) vs. SVD when fine-tuning OPT-1.3B/13B on SST-2. The runtime is further decomposed into three components: sampling the projection matrix $\mathbf{P}$, forward passes for ZO gradient estimation, and the msign computation. As shown, using SVD for exact msign computation indeed improves accuracy over the NS-based implementation. However, SVD incurs approximately $3\times$ higher runtime than NS for the matrix-sign operation. This substantial efficiency gap is the primary reason we adopt NS in ZO-Muon.

Additional results. We report additional validations in **Appendix I**. **Fig. A3** shows the fast convergence of ZO-Muon across multiple datasets, complementing Fig. 1. **Tab. A2** reports results on Gemma2-2B [48], where ZO-Muon again consistently outperforms other ZO methods, in line with Tab. 1. **Fig. A4** reports the performance of ZO-Muon and the baselines with matched ranks. We also study the resampling frequency of the projection matrix $\mathbf{P}$ and show that lazy resampling stabilizes ZO training in **Fig. A5**.

6 Conclusion

In this paper, we propose ZO-Muon, a novel ZO optimization method that bridges the gap between memory efficiency and convergence speed in deep model training. By combining subspace optimization with spectral gradient orthogonalization, ZO-Muon resolves the fundamental variance–efficiency tension in standard ZO methods. Extensive experiments on LLMs and ViTs show that ZO-Muon consistently outperforms state-of-the-art baselines in both performance and efficiency. We refer readers to Appendix J-L for discussions on limitations, ethics statement, and LLM usage.

Acknowledgment

The work of Y. Lang, C. Wang, Y. Zhang, and S. Liu was supported in part by the National Science Foundation (NSF) CAREER Award IIS-2338068. Z. Zhang was supported by NSF CCF-2419889. The authors declare no competing financial interests related to this work beyond the funding disclosed above.

References

[1] 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.

[2] 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.

[3] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

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

[6] Yiming Chen, Yuan Zhang, Yin Liu, Kun Yuan, and Zaiwen Wen. A memory efficient randomized subspace optimization method for training large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=XuCf87V8OF`.

[7] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning:

Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

[8] 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.

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

[10] 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 *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Vota6rFhBQ.

[11] 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. In *International Conference on Machine Learning*, pages 59173–59190. PMLR, 2024.

[12] Qitao Tan, Jun Liu, Zheng Zhan, Caiwen Ding, Yanzhi Wang, Xiaolong Ma, Jaewoo Lee, Jin Lu, and Geng Yuan. Harmony in divergence: Towards fast, accurate, and memory-efficient zeroth-order LLM fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Rx6m16By6l.

[13] 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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=TjwOACu3NL.

[14] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

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

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

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

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

[19] 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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bEqI61iBue.

[20] 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*, pages 288–297. PMLR, 2018.

[21] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

[22] Hanzhen Zhao, Shihong Ding, Cong Fang, and Zhouchen Lin. PaZO: Preconditioned accelerated zeroth-order optimization for fine-tuning LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=b2IU6QOOfo`.

[23] Shaocong Ma and Heng Huang. Revisiting zeroth-order optimization: Minimum-variance two-point estimators and directionally aligned perturbations. In *The Thirteenth International Conference on Learning Representations*, 2025.

[24] Wentao Guo, Jikai Long, Yimeng Zeng, Zirui Liu, Xinyu Yang, Yide Ran, Jacob R Gardner, Osbert Bastani, Christopher De Sa, Xiaodong Yu, et al. Zeroth-order fine-tuning of llms with extreme sparsity. *arXiv preprint arXiv:2406.02913*, 2024.

[25] 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*, pages 4475–4485, 2025.

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

[27] Sizhe Dang, Yangyang Guo, Yanjun Zhao, Haishan Ye, Xiaodong Zheng, Guang Dai, and Ivor Tsang. Fzoo: Fast zeroth-order optimizer for fine-tuning large language models towards adam-scale speed. *arXiv preprint arXiv:2506.09034*, 2025.

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

[29] Yuezhang Peng, Yuxin Liu, Fei Wen, and Xie Chen. Muzo: Leveraging multiple queries and momentum for zeroth-order fine-tuning of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8577–8595, 2025.

[30] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL `https://kellerjordan.github.io/posts/muon/`.

[31] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

[32] Egor Petrov, Grigoriy Evseev, Aleksey Antonov, Andrey Veprikov, Nikolay Bushkov, Stanislav Moiseev, and Aleksandr Beznosikov. Leveraging coordinate momentum in signsgd and muon: Memory-optimized zero-order. *arXiv preprint arXiv:2506.04430*, 2025.

[33] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 7319–7328, 2021.

[34] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

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

[36] Prakhar Kaushik, Shravan Chaudhari, Ankit Vaidya, Rama Chellappa, and Alan Yuille. The universal weight subspace hypothesis. *arXiv preprint arXiv:2512.05117*, 2025.

[37] Ryota Nozawa, Pierre-Louis Poirion, and Akiko Takeda. Zeroth-order random subspace algorithm for non-smooth convex optimization. *Journal of Optimization Theory and Applications*, 204(3):53, 2025.

[38] David Kozak, Cesare Molinari, Lorenzo Rosasco, Luis Tenorio, and Silvia Villa. Zeroth-order optimization with orthogonal random directions. *Mathematical Programming*, 199(1): 1179–1219, 2023.

[39] Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. In *OPT 2024: Optimization for Machine Learning*, 2024. URL `https://openreview.net/forum?id=ux18f5nOpD`.

[40] Ahmed Khaled, Kaan Ozkara, Tao Yu, Mingyi Hong, and Youngsuk Park. Muonbp: Faster muon via block-periodic orthogonalization. *arXiv preprint arXiv:2510.16981*, 2025.

[41] Chuan He, Zhanwang Deng, and Zhaosong Lu. Low-rank orthogonalization for large-scale matrix optimization with applications to foundation model training. *arXiv preprint arXiv:2509.11983*, 2025.

[42] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53 (2):217–288, 2011.

[43] 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.

[44] 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.

[45] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

[46] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[47] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[48] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

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

[50] 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.

[51] Minhak Song, Liang Zhang, Bingcong Li, Niao He, Michael Muehlebach, and Sewoong Oh. Zeroth-order optimization at the edge of stability. *arXiv preprint arXiv:2604.14669*, 2026.

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

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

[54] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, JinFeng Yi, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJlk6iRqKX.

[55] Sijia Liu, Songtao Lu, Xiangyi Chen, Yao Feng, Kaidi Xu, Abdullah Al-Dujaili, Mingyi Hong, and Una-May O'Reilly. Min-max optimization without gradients: Convergence and applications to black-box evasion and poisoning attacks. In *International conference on machine learning*, pages 6282–6293. PMLR, 2020.

[56] Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jinfeng Yi, Mingyi Hong, Shiyu Chang, and Sijia Liu. How to robustify black-box ML models? a zeroth-order optimization perspective. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=W9G_ImpHlQd.

[57] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022.

[58] Wenyang Hu, Yao Shu, Zongmin Yu, Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, See-Kiong Ng, and Bryan Kian Hsiang Low. Localized zeroth-order prompt optimization. *Advances in Neural Information Processing Systems*, 37:86309–86345, 2024.

[59] Haozhen Zhang, Zhaogeng Liu, Bin Gu, and Yi Chang. Leveraging first and zeroth-order gradient to address imbalanced black-box prompt tuning via minimax optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22407–22415, 2025.

[60] Jiaqi Gu, Chenghao Feng, Zheng Zhao, Zhoufeng Ying, Ray T Chen, and David Z Pan. Efficient on-chip learning for optical neural networks through power-aware sparse zeroth-order optimization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7583–7591, 2021.

[61] Pengyu Zhang, Yingjie Liu, Yingbo Zhou, Xian Wei, and Mingsong Chen. When foresight pruning meets zeroth-order optimization: Efficient federated learning for low-memory devices. *Journal of Systems Architecture*, 168:103526, 2025.

[62] Qitao Tan, Sung-En Chang, Rui Xia, Huidong Ji, Chence Yang, Ci Zhang, Jun Liu, Zheng Zhan, Zhenman Fang, Zhuo Zou, et al. Perturbation-efficient zeroth-order optimization for hardware-friendly on-device training. In *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9. IEEE, 2025.

[63] Hiroshi Sawada, Kazuo Aoyama, and Yuya Hikima. Natural perturbations for black-box training of neural networks by zeroth-order optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ULAQ9GmJlo.

[64] Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=uubBZKM99Y.

[65] Yutong He, Pengrui Li, Yipeng Hu, Chuyan Chen, and Kun Yuan. Subspace optimization for large language models with convergence guarantees. *arXiv preprint arXiv:2410.11289*, 2024.

[66] Kaizhao Liang, Bo Liu, Lizhang Chen, and Qiang Liu. Memory-efficient llm training with online subspace descent. *Advances in Neural Information Processing Systems*, 37:64412–64432, 2024.

[67] Haochen Zhang, Junze Yin, Guanchu Wang, Zirui Liu, Tianyi Zhang, Anshumali Shrivastava, Lin Yang, and Vladimir Braverman. I3s: Importance sampling subspace selection for low-rank optimization in llm pretraining. *arXiv preprint arXiv:2502.05790*, 2025.

[68] Yehonathan Refael, Guy Smorodinsky, Tom Tirer, and Ofir Lindenbaum. SUMO: Subspace-aware moment-orthogonalization for accelerating memory-efficient LLM training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=DIjRvEKOeG.

[69] 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.

[70] Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4823–4836, 2025.

[71] Kai Lion, Liang Zhang, Bingcong Li, and Niao He. PoLAR: Polar-decomposed low-rank adapter representation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=jDxFD45kkc.

[72] Muqing Liu, Chongjie Si, and Yuheng Jia. FlexloRA: Entropy-guided flexible low-rank adaptation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=tqnkbdYWWm.

[73] Chao Ma, Wenbo Gong, Meyer Scetbon, and Edward Meeds. SWAN: SGD with normalization and whitening enables stateless LLM training. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=CQZXGmw5vO.

[74] Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making muon & scion great again!(bridging theory and practice of lmo-based optimizers for llms). *arXiv preprint arXiv:2505.13416*, 2025.

[75] Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. Dion: Distributed orthonormalized updates. *arXiv preprint arXiv:2504.05295*, 2025.

[76] Noah Amsel, David Persson, Christopher Musco, and Robert M Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm. *arXiv preprint arXiv:2505.16932*, 2025.

[77] Wenbo Gong, Javier Zazo, Qijun Luo, Puqian Wang, James Hensman, and Chao Ma. Aro: A new lens on matrix optimization for large models. *arXiv preprint arXiv:2602.09006*, 2026.

[78] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[79] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer, 2005.

[80] Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124, 2019.

[81] 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)*, pages 2924–2936, 2019.

[82] Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. 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)*, pages 1267–1273, 2019.

[83] 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*, pages 2383–2392, 2016.

[84] Yehonathan Refael, Jonathan Svirsky, Boris Shustin, Wasim Huleihel, and Ofir Lindenbaum. Adarankgrad: Adaptive gradient rank and moments for memory-efficient LLMs training and fine-tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LvNROciCne.

Appendix

A Detailed Related Work

ZO optimization and applications in deep models. ZO optimization [7–9, 49–51] estimates gradients via finite differences of function values without computing FO derivatives. As detailed in Sec. 1 and thus omitted here for brevity, a large body of foundational work on ZO optimization has focused on reducing estimation variance to improve accuracy and the trade-off with query efficiency. Empirically, the effectiveness of ZO methods has been demonstrated in solving small-scale problems, such as input-level black-box adversarial attack and defense [14, 15, 52–56], input-level prompt design for foundation models [57–59], and resource-limited on-device learning [60–62]. More recently, ZO optimization has expanded to large-scale deep model training, including black-box or non-differentiable settings [17, 63] and memory-efficient fine-tuning [10–13]. Focusing on the latter, to the best of our knowledge, this is the first work to demonstrate a *win–win* improvement in both accuracy and efficiency via subspace gradient orthogonalization.

Subspace optimization for efficient training. Since the weights or gradients in deep model training often exhibit low-rank structure [33–36, 64], a growing body of work has explored training in low-dimensional subspaces to improve memory efficiency [65–68]. Notably, LoRA-style methods [34, 69–72] directly constrain trainable parameters to low-rank matrices, but can be less effective in some settings due to the limited expressiveness of low-rank parameterizations [64] and insufficient directional diversity [71]. Subspace perspectives have been explored in ZO optimization. Prior work [37, 38] develops theoretical subspace formulations but does not address practical large-scale fine-tuning, while more practical methods like LOZO [18] and SubZero [25] use low-dimensional Gaussian matrices for perturbation and updates. As we show later, our method clearly outperforms these approaches by exploiting the variance-reduction benefits of subspace ZO optimization in the spectral domain.

Emerging gradient-orthogonalized methods for deep model training. Recent years have witnessed growing interest in gradient-orthogonalized optimization methods for training large-scale deep models, particularly foundation models, driven by the need to exploit the matrix-structured and spectral properties of gradients that are poorly captured by conventional vector-based optimizers. The Muon optimizer [30] and its subsequent variants [31, 40, 41, 73–77] embody this shift by performing *gradient orthogonalization* (**GO**), which projects gradient updates onto a spectrally constrained set via matrix whitening and can be interpreted as a steepest-descent step under a spectral-norm constraint [39]. This spectral treatment leads to faster convergence and reduced optimizer-state memory compared to Adam, enabling Muon to scale to foundation-model training, most notably in Moonlight [31], a 3B/16B Mixture-of-Experts (MoE) model. Despite this progress, GO has remained largely confined to *FO pre-training*, and extending it to ZO settings is nontrivial due to the lower quality of ZO gradient estimates. JAGUAR-Muon [32] attempts to apply GO to full-space ZO momentum, but incurs additional memory and runtime overhead over MeZO [10] and shows very limited performance gain. This is likely because applying GO to noisy, high-dimensional ZO estimates amplifies non-informative spectral components, a limitation also confirmed by our experiments. By contrast, we show that GO becomes effective for ZO optimization *only when coupled with a subspace formulation*, *i.e.*, *subspace gradient orthogonalization*.

B Intrinsic Rank Measurement

To estimate the intrinsic dimensionality of the gradients, we perform Singular Value Decomposition (SVD) on the gradient matrix $\mathbf{G}$. Let $\{\sigma_i\}_{i=1}^{n}$ denote the singular values of $\mathbf{G}$ sorted in descending order. We determine the effective rank by identifying the number of principal components k required to capture 99.99% of the gradient energy. This relationship is formulated as finding the cutoff k satisfying: $\mathcal{E}(k) = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{n} \sigma_i^2} \geq 0.9999$.

C Ineffectiveness of Full Space Gradient Orthogonalization

Complementary to **Fig. 5**, we additionally validate the *ineffectiveness* of full space ZO gradient orthogonalization by fine-tuning OPT-1.3B on SST-2, shown in **Fig. A1**. With the same query

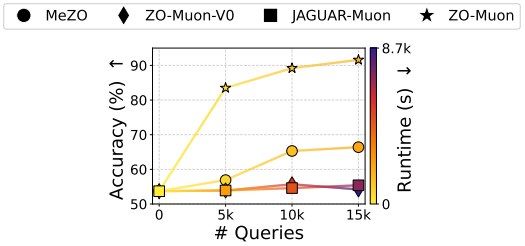


Figure A1: Fine-tune accuracy and runtime of full space gradient orthogonalization methods JAGUAR-Muon and ZO-Muon-V0 compared to MeZO and ZO-Muon, demonstrated by fine-tuning OPT-1.3B on SST-2.

number, ZO-Muon-V0 and JAGUAR-Muon [32] does not outperform MeZO in fine-tune accuracy. Additionally, they consume more runtime compared to MeZO. In contrast, the *subspace* gradient orthogonalization ZO-Muon outperforms the above methods in accuracy and does not sacrifice runtime efficiency.

D ZO-Muon Algorithm

Algorithm 1 details our proposed ZO-Muon method. ZO-Muon treats parameters as matrices and optimizes layer-wise matrices $\{\mathbf{X}_l\}$ where l represents each layer. First, the projection matrices $\{\mathbf{P}_l\}$ are lazily sampled every interval v, corresponding to "*subspace sampling*". Second, we perform *Subspace-RGE* following (5), making $N_q > 1$ queries and performing a forward-difference estimate, and average the subspace ZO estimates. Finally, shown in "full-space weight update", we follow (10) to orthogonalize the subspace gradient and project them back to full-space for model update.

Algorithm 1 ZO-Muon

Input: Parameters $\mathbf{X} = \{\mathbf{X}_l\}$, (where $\mathbf{X}_l \in \mathbb{R}^{m_l \times n_l}$), loss function $f(\mathbf{X})$, projection matrices $\mathbf{P} = \{\mathbf{P}_l\}$, learning rate η, perturbation scale μ, rank parameter r, query number N_q, projection matrix resample interval v, train steps T.
// Initialize Projection Matrices
Sample column-orthogonal projection matrices $\{\mathbf{P}_l\}$ via QR decomposition.
for $t = 0, \ldots, T-1$ **do**
  // 1. Subspace Sampling (Periodic Update)
  if $t > 0$ **and** $t \pmod v = 0$ **then**
    Resample $\{\mathbf{P}_l\}$ via QR decomposition.
  end if
  // 2. Subspace-RGE
  Compute subspace ZO gradient estimate via (5) with $N_q > 1$.
  // 3. Full-space weight update
  Perform subspace gradient orthogonalization, then project back to full space to update via (10).
end for
Return $\mathbf{X}$;

E Proof of Proposition 1

Proof: With $\mathrm{msign}(\mathbf{G}) = \mathbf{G}(\mathbf{G}^\top\mathbf{G})^{-\frac{1}{2}}$ and using $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$, we obtain

$$\begin{aligned}\mathbf{P}\,\mathrm{msign}(\mathbf{P}^\top\mathbf{G}) &= \mathbf{P}\,\mathbf{P}^\top\mathbf{G}\big((\mathbf{P}^\top\mathbf{G})^\top(\mathbf{P}^\top\mathbf{G})\big)^{-1/2} \\ &= \mathbf{P}\,\mathbf{P}^\top\mathbf{G}\big(\mathbf{G}^\top\mathbf{P}\mathbf{P}^\top\mathbf{G}\big)^{-1/2}. \end{aligned} \tag{A1}$$

Next, the SVD of $\mathbf{G}$ yields

$$\mathbf{G} = \mathbf{U}_{[:,:k]}\,\mathrm{diag}(\boldsymbol{\sigma})\,\mathbf{V}_{[:,:k]}^\top,$$

where $\mathrm{diag}(\boldsymbol{\sigma}) \in \mathbb{R}^{k\times k}$ is the diagonal matrix containing the nonzero singular values of $\mathbf{G}$, and $k = \mathrm{rank}(\mathbf{G})$.

Since $\mathbf{P} = \mathbf{U}_{[:,:k]}$ and $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$, we have

$$\begin{aligned}\mathbf{P}\mathbf{P}^\top\mathbf{G} &= \mathbf{U}_{[:,:k]}\mathbf{U}_{[:,:k]}^\top\mathbf{U}_{[:,:k]}\,\mathrm{diag}(\boldsymbol{\sigma})\,\mathbf{V}_{[:,:k]}^\top \\ &= \mathbf{U}_{[:,:k]}\,\mathrm{diag}(\boldsymbol{\sigma})\,\mathbf{V}_{[:,:k]}^\top \\ &= \mathbf{G}. \end{aligned} \tag{A2}$$

That is, $\mathbf{P}\mathbf{P}^\top$ acts as the identity on the column space of $\mathbf{G}$. Substituting (A2) into (A1), we finally obtain $\mathbf{P}\mathrm{msign}(\mathbf{P}^\top\mathbf{G}) = \mathrm{msign}(\mathbf{G})$. This completes the proof.

F Convergence Analysis

To simplify the analysis without loss of generality, for the matrix parameter $\mathbf{X} \in \mathbb{R}^{m\times n}$, we assume a static column-orthogonal projection matrix $\mathbf{P} \in \mathbb{R}^{m\times r}$ ($\mathbf{P}^\top\mathbf{P} = \mathbf{I}$) with $r \ll \min\{m, n\}$. The exact subspace gradient is denoted as $\nabla_{\mathbf{Z}} f(\mathbf{X}) = \mathbf{P}^\top\nabla f(\mathbf{X})$. ZO-Muon utilizes randomized gradient estimate (RGE) to compute the subspace gradient using N_{q} queries:

$$\hat{\nabla}_{\mathbf{Z}} f(\mathbf{X}) = \frac{1}{N_{\mathrm{q}}}\sum_{i=1}^{N_{\mathrm{q}}} \frac{f(\mathbf{X} + \mu\mathbf{P}\boldsymbol{\Psi}_i) - f(\mathbf{X})}{\mu}\,\boldsymbol{\Psi}_i. \tag{A3}$$

The above yields an unbiased gradient estimate of the *smoothed* objective $f_\mu(\mathbf{X}) = \mathbb{E}_{\boldsymbol{\Psi}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[f(\mathbf{X} + \mu\mathbf{P}\boldsymbol{\Psi})]$, where $\mu > 0$ is the smoothing parameter. That is,

$$\mathbb{E}_{\boldsymbol{\Psi}}[\hat{\nabla}_{\mathbf{Z}} f(\mathbf{X})] = \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}). \tag{A4}$$

The above unbiasedness property facilitates convergence analysis of ZO-Muon in expectation. Thus, we focus on the convergence rate of its smoothed variant, which, under the unbiasedness condition, induces the following Muon update:

$$\mathbf{X}_{t+1} = \mathbf{X}_t - \eta\mathbf{P}\mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)), \tag{A5}$$

where t represents the optimization step.

F.1 Assumptions

We make the following assumptions standard for the convergence analysis of non-convex optimization.

Assumption 1 (L-smoothness). The objective function f is smooth with constant L, meaning that for any $\mathbf{X}, \mathbf{Y}$, it satisfies:

$$f(\mathbf{Y}) \le f(\mathbf{X}) + \langle\nabla f(\mathbf{X}), \mathbf{Y} - \mathbf{X}\rangle + \frac{L}{2}\|\mathbf{Y} - \mathbf{X}\|_{\mathrm{F}}^2, \tag{A6}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm.

Assumption 2 (Subspace alignment). The projection matrix $\mathbf{P} \in \mathbb{R}^{m\times r}$ captures a fraction $\delta \in (0, 1]$ of the full gradient. Specifically, $\|\nabla_{\mathbf{Z}} f(\mathbf{X})\|_{\mathrm{F}} \ge \delta\|\nabla f(\mathbf{X})\|_{\mathrm{F}}$.

F.2 Supporting Lemmas

Lemma 1 (Theorem 1 in [9]) *Under Assumption 1, the expected smoothing bias of the ZO subspace gradient estimator is bounded by:*

$$\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_{\mathrm{F}} \le \mu L\sqrt{rn}, \tag{A7}$$

where t denotes the optimization step, μ is the smoothing parameter, L is the Lipschitz constant defined in Assumption 1, r is the projection dimension, and n is the column dimension of $\mathbf{X}$.

Lemma 2 *For the parameter update driven by (A5), the inner product between the subspace gradient and the gradient-orthogonalized direction satisfies:*

$$\langle\nabla_{\mathbf{Z}} f(\mathbf{X}_t), \mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\rangle \ge \|\nabla_{\mathbf{Z}} f(\mathbf{X}_t)\|_* - 2\sqrt{r}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_{\mathrm{F}}, \tag{A8}$$

where $\|\cdot\|_*$ represents the nuclear norm of a matrix.

Proof. Because $\mathbf{P} \in \mathbb{R}^{m\times r}$ is column-orthogonal ($\mathbf{P}^\top\mathbf{P} = \mathbf{I}$), the Frobenius norm is invariant under projection, meaning $\|\mathbf{P}\mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\|_{\mathrm{F}}^2 = \|\mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\|_{\mathrm{F}}^2$. Furthermore, because the subspace gradient $\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t) \in \mathbb{R}^{r\times n}$ has a maximum rank of r, and the msign operator sets all non-zero singular values exactly to 1, the squared Frobenius norm, which is the sum of squared singular values, is bounded by r. Thus, $\|\mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\|_{\mathrm{F}}^2 \le r$. Therefore:

$$\begin{aligned}\langle \nabla_{\mathbf{Z}} f(\mathbf{X}_t), \mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\rangle &= \langle \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t), \mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\rangle \\ &\quad + \langle \nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t), \mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\rangle \\ &\ge \|\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_* - \|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_{\mathrm{F}} \|\mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\|_{\mathrm{F}} \\ &\ge \|\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_* - \sqrt{r}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_{\mathrm{F}}.\end{aligned}$$

Applying the reverse triangle inequality for nuclear norms ($\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t)\|_* \le \|\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_* + \|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_*$) and leveraging the fact that $\|\mathbf{A}\|_* \le \sqrt{r}\|\mathbf{A}\|_{\mathrm{F}}$ for rank r matrices, we obtain the stated lower bound (A8). □

F.3 Main Convergence Result

Theorem 1 *Suppose Assumptions 1 and 2 hold. Let the learning rate be chosen as $\eta = \sqrt{\frac{2\Delta}{LrT}}$ where $\Delta = f(\mathbf{X}_0) - f^*$ is the initial suboptimality, and f^* is the optimal value. Let the smoothing parameter be set as $\mu = \frac{1}{L\sqrt{rnT}}$. Then, the iterates generated by the smoothed ZO-Muon (A5) converge to a stationary point at a rate of:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(\mathbf{X}_t)\|_{\mathrm{F}} \le \mathcal{O}\left(\frac{1}{\delta}\sqrt{\frac{r}{T}}\right), \tag{A9}$$

where T denotes the number of iterations, δ is the subspace alignment ratio, and r is the projection dimension.

Proof. By the L-smoothness assumption (Assumption 1), the parameter update (A5) yields:

$$\begin{aligned}f(\mathbf{X}_{t+1}) &\le f(\mathbf{X}_t) - \eta\langle \nabla f(\mathbf{X}_t), \mathbf{P}\mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\rangle + \frac{L\eta^2}{2}\|\mathbf{P}\mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\|_{\mathrm{F}}^2 \\ &\le f(\mathbf{X}_t) - \eta\langle \nabla_{\mathbf{Z}} f(\mathbf{X}_t), \mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\rangle + \frac{L\eta^2 r}{2},\end{aligned}$$

where the last inequality holds due to $\|\mathbf{P}\mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\|_{\mathrm{F}}^2 = \|\mathrm{msign}(\nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t))\|_{\mathrm{F}}^2 \le r$ as used in the proof of Lemma 2.

Applying the inner product bound from Lemma 2, we obtain

$$f(\mathbf{X}_{t+1}) \le f(\mathbf{X}_t) - \eta\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t)\|_* + 2\eta\sqrt{r}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_{\mathrm{F}} + \frac{L\eta^2 r}{2}.$$

Rearranging the terms to isolate the gradient norm and summing over T iterations, we obtain a telescoping sum for the objective function values:

$$\begin{aligned}\eta\sum_{t=0}^{T-1}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t)\|_* &\le \sum_{t=0}^{T-1}[f(\mathbf{X}_t) - f(\mathbf{X}_{t+1})] + 2\eta\sqrt{r}\sum_{t=0}^{T-1}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_{\mathrm{F}} + \frac{TL\eta^2 r}{2} \\ &= f(\mathbf{X}_0) - f(\mathbf{X}_T) + 2\eta\sqrt{r}\sum_{t=0}^{T-1}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_{\mathrm{F}} + \frac{TL\eta^2 r}{2}.\end{aligned}$$

By the definition of f^*, we obtain $f(\mathbf{X}_0) - f(\mathbf{X}_T) \le f(\mathbf{X}_0) - f^* := \Delta$. Substituting this yields:

$$\eta\sum_{t=0}^{T-1}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t)\|_* \le \Delta + 2\eta\sqrt{r}\sum_{t=0}^{T-1}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_{\mathrm{F}} + \frac{TL\eta^2 r}{2}.$$

Further, dividing the entire equation by ηT, we obtain:

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t)\|_* \le \frac{\Delta}{\eta T} + \frac{L\eta r}{2} + \frac{2\sqrt{r}}{T}\sum_{t=0}^{T-1}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t) - \nabla_{\mathbf{Z}} f_\mu(\mathbf{X}_t)\|_{\mathrm{F}}.$$

Substituting (A7) from Lemma 1, we obtain:

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t)\|_* \leq \frac{\Delta}{\eta T} + \frac{L\eta r}{2} + 2\mu L r\sqrt{n}.$$

Since the nuclear norm is bounded below by the Frobenius norm ($\|\mathbf{A}\|_* \geq \|\mathbf{A}\|_{\mathrm{F}}$), we apply the subspace alignment condition from Assumption 2 ($\|\nabla_{\mathbf{Z}} f(\mathbf{X}_t)\|_{\mathrm{F}} \geq \delta\|\nabla f(\mathbf{X}_t)\|_{\mathrm{F}}$) to convert this to the full gradient:

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(\mathbf{X}_t)\|_{\mathrm{F}} \leq \frac{1}{\delta}\left[\frac{\Delta}{\eta T} + \frac{L\eta r}{2} + 2\mu L r\sqrt{n}\right]. \tag{A10}$$

Equation (A10) represents the generalized non-asymptotic convergence bound. To obtain the asymptotic rate, we set $\eta = \sqrt{\frac{2\Delta}{LrT}}$ and $\mu = \frac{1}{L\sqrt{rnT}}$.

Evaluating the learning rate terms:

$$\frac{\Delta}{\eta T} + \frac{L\eta r}{2} = \sqrt{\frac{\Delta L r}{2T}} + \sqrt{\frac{\Delta L r}{2T}} = \sqrt{\frac{2\Delta L r}{T}} = \mathcal{O}\left(\sqrt{\frac{r}{T}}\right).$$

Evaluating the smoothing bias term:

$$2\mu L r\sqrt{n} = 2\left(\frac{1}{L\sqrt{rnT}}\right) Lr\sqrt{n} = 2\sqrt{\frac{r}{T}} = \mathcal{O}\left(\sqrt{\frac{r}{T}}\right).$$

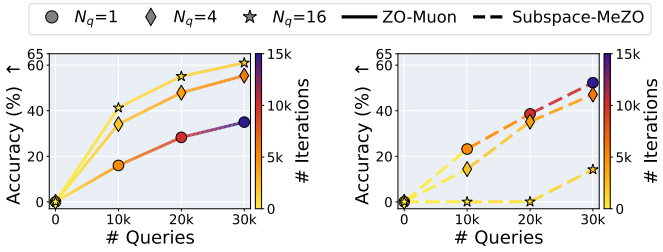
Combining these simplified components confirms the final asymptotic rate is independent of n:

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(\mathbf{X}_t)\|_{\mathrm{F}} \leq \mathcal{O}\left(\frac{1}{\delta}\sqrt{\frac{r}{T}}\right).$$

The proof is now complete. □

G Hyperparameter Study on N_q in ZO-Muon

In **Fig. A2**, we compare ZO-Muon with Subspace-MeZO under different query number N_q, plotting performance against the total number of queries used during training. Shown in Fig. A2(a), there exists a multi-query paradox: Subspace-MeZO with $N_q > 1$ performs worse than $N_q = 1$. On the contrary, ZO-Muon does not suffer from this issue (*i.e.*, additional queries do not lead to diminishing optimization returns). Moreover, as shown in Fig. A2(b), although ZO-Muon uses $N_q = 4$, this does not hurt query efficiency in practice compared to Subspace MeZO with $N_q = 1$, since ZO-Muon converges in significantly fewer iterations.



(a) ZO-Muon (b) Subspace-MeZO

Figure A2: Fine-tuning accuracy vs. query cost for (a) ZO-Muon and (b) Subspace-MeZO when fine-tuning ViT-B on CIFAR-100, with marker color indicating the used iteration number. The per-iteration query budget is set to $N_q \in \{1, 4, 16\}$. *(a)* ZO-Muon benefits from increasing N_q, with a notable performance gain when moving from $N_q = 1$ to $N_q = 4$. *(b)* Under the same setting, Subspace-MeZO exhibits a multi-query paradox, where performance degrades when $N_q > 1$.

H Detailed Experiment Setups

Models and datasets. For LLM experiments, we evaluate Gemma2-2B [48], LlaMA3-8B [43] and OPT-13B [2] on the SuperGLUE benchmark [44], including the tasks SST-2 [78], RTE [79], CB [80],

BoolQ [81], WiC [82] and SQuAD [83]. Following previous literature [10, 18, 19, 25], we randomly sample 1,000 train samples and 1,000 test samples. We adopt the same prompts as MeZO [10].

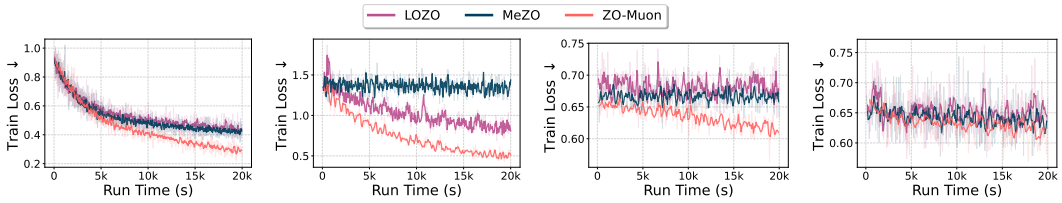
For ViT experiments, we evaluate ViT-B/16 and ViT-L/16 [45] fine-tuned on the CIFAR datasets [46]. Data processing and evaluations are based on the well-established ViT pytorch code base[1].

Implementation details. Our ZO-Muon, as well as baselines LOZO and SubZero treat model parameters as *matrices* instead of vectors. Following Muon [30], we flatten the last three dimensions of 4D convolutional parameters and treat them as 2D matrices for optimization. Vector parameters, as well as the input and output layers, are optimized with MeZO.

Hyperparameters. We adopt hyperparameter settings consistent with existing literature to ensure rigorous benchmarking. For LLMs, we use a batch size of 16 and train for 20,000 steps, utilizing a constant learning rate without weight decay [10, 18, 25]. The perturbation step size is fixed at $\mu = 10^{-3}$ based on its demonstrated robustness across tasks [10, 18]. For ViT experiments, we set the batch size to 64 (CIFAR-10) and 256 (CIFAR-100). The necessity for increased batch sizes on CIFAR-100 is demonstrated in Fig. 7 and Sec. 5. We also fix the perturbation stepsize $\mu = 10^{-3}$. For algorithmic hyperparameters (e.g., sparsity for S-MeZO; rank r and interval v for subspace-based methods like LOZO and SubZero), we strictly follow the original baseline implementations. For LoRA, we follow previous works [10, 18] and set rank $r = 8$ and the scaling factor $\alpha = 16$. Detailed settings for both ZO and FO methods are listed in **Tab. A1**.

Computing resources. Experiments are primarily conducted on A6000 GPUs with 48GB memory. OPT-13B full Adam fine-tuning is run on H100 GPUs with 100GB memory since A6000 does not have enough memory under our setting.

I Additional Experiment Results



(a) SST-2 (b) SQuAD (c) RTE (d) BoolQ

Figure A3: ZO-Muon achieves faster convergence rate compared to MeZO and LOZO, demonstrated through fine-tuning OPT-13B on SST-2, SQuAD, RTE and BoolQ.

Loss curves of LLM fine-tuning. Complementary to Fig. 1, **Fig. A3** shows the training loss curves of ZO-Muon and representative baselines (MeZO and LOZO) for OPT-13B fine-tuning on four downstream tasks. Note that each method is plotted using its best hyperparameter setting from the search grid (see Appendix H). Across all four tasks, ZO-Muon achieves the fastest convergence among the baselines.

Gemma2-2B results. **Tab. A2** presents fine-tuning results on Gemma2-2B. Consistent with Sec. 5.2, ZO-Muon achieves the best overall ZO performance and comes closest to FO methods. Notably, it improves over MeZO by 10.2% on RTE and 8.5% on BoolQ.

Performance comparison with matched rank. In the main experiments, LOZO and SubZero use the ranks suggested in their original papers, and all methods share the same overall query budget. To isolate the effect of rank, **Fig. A4** compares LOZO, SubZero and ZO-Muon at the same ranks $r \in \{32, 64, 128\}$ with the same budget of 40k queries on OPT-1.3B / SST-2. ZO-Muon outperforms both baselines at every rank. The gain therefore comes from combining subspace optimization with Muon-style gradient orthogonalization, not from a larger rank. This agrees with the analysis of Subspace RGE in Fig. 6, where a larger rank increases the variance of the gradient estimate.

[1] `https://github.com/jeonsworld/ViT-pytorch`

Table A1: Hyperparameter settings of ZO and FO methods used for fine-tuning LLMs and ViTs.

Methods	**Hyperparameters**	**Values**	
		LLM	**ViT**
MeZO	Learning rate	$\{1e-7, 1e-6\}$ or $\{1e-7, 5e-7, 1e-6\}$ for SQuAD	$\{1e-5, 1e-4\}$
	Train steps	20k	20k
S-MeZO	Learning rate	$\{1e-7, 1e-6\}$ or $\{1e-7, 5e-7, 1e-6\}$ for SQuAD	$\{1e-5, 1e-4\}$
	Sparsity	0.8	0.8
	Train steps	20k	20k
HiZOO	Learning rate	$\{1e-7, 1e-6\}$ or $\{1e-7, 5e-7, 1e-6\}$ for SQuAD	$\{1e-5, 1e-4\}$
	Train steps	20k	20k
LOZO	Learning rate	$\{1e-7, 1e-6\}$	$\{1e-5, 1e-4\}$
	Rank (r)	$\{2, 4, 8\}$	$\{2, 4, 8\}$
	Interval (v)	100	100
	Train steps	20k	20k
SubZero	Learning rate	$\{1e-7, 1e-6\}$	$\{1e-5, 1e-4\}$
	Rank (r)	$\{24, 48\}$	$\{24, 48\}$
	Interval (v)	1000	1000
	Train steps	20k	20k
Subspace-MeZO	Learning rate	$1e-5$	$\{1e-5, 1e-4\}$
	Rank (r)	$\{64, 128\}$	$\{64, 128\}$
	Interval (v)	100	100
	Train steps	20k	20k
ZO-Muon	Learning rate	$1e-2$	$1e-2$
	Rank (r)	$\{64, 128\}$	$\{64, 128\}$
	Interval (v)	100	100
	Train steps ($N_q = 4$)	8k	10k
	Train steps ($N_q = 8$)	4k	–
	Train steps ($N_q = 16$)	–	2k
Full	Learning rate	$\{5e-6, 1e-5\}$	$\{1e-3, 5e-3\}$
	Epochs	10	10
LoRA	Learning rate	$\{5e-6, 1e-5\}$	$\{1e-3, 5e-3\}$
	Epochs	10	10
	Rank (r)	8	8
	Alpha (α)	16	16
Galore	Learning rate	$\{5e-6, 1e-5\}$	$\{1e-3, 5e-3\}$
	Epochs	10	10
	Rank (r)	128	128

Projection matrix resampling interval Recall that in ZO-Muon we *lazily* resample the subspace projection matrix $\mathbf{P}$ every 100 steps (denoted by the resampling interval $v = 100$). To validate this design choice, **Fig. A5** compares ZO-Muon with different resampling intervals $v \in \{1, 100, 500\}$. As shown, overly frequent resampling does not improve performance on either LLM or ViT fine-tuning. The case of $v = 1$ consistently underperforms $v = 100$, while $v = 500$ achieves performance comparable to $v = 100$. This is because frequent subspace switching (*i.e.*, resampling $\mathbf{P}$) prevents

Table A2: Performance of ZO-Muon and baselines on Gemma2-2B under the same settings as Tab.1.

Method	**Gemma2-2B Fine-tuning**					
	SST-2	**RTE**	**CB**	**BoolQ**	**WiC**	**SQuAD**
• Full	94.0	84.0	84.0	90.0	72.0	89.4
• LoRA	91.6	57.8	55.4	75.2	56.1	86.9
• Galore	94.6	79.1	66.1	80.4	64.1	87.3
MeZO	91.7	53.0	60.7	66.1	57.1	74.1
S-MeZO	92.7	53.8	53.6	63.9	55.8	76.1
HiZOO	92.1	53.1	60.7	66.6	55.2	73.4
LOZO	92.2	56.3	57.1	65.0	50.2	77.3
SubZero	91.7	56.7	**67.8**	63.5	58.3	72.1
FZOO	93.6	57.0	64.3	66.2	55.5	83.2
ZO-Muon	**94.2**	**63.2**	64.3	**74.6**	**60.3**	**84.3**

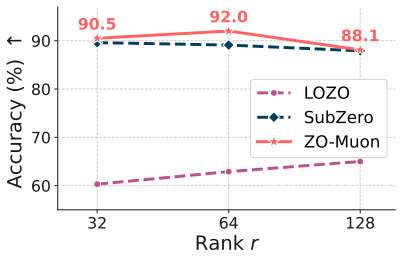
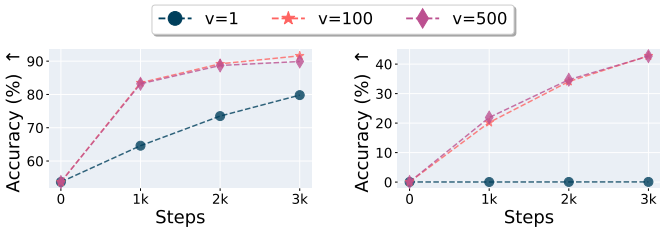


Figure A4: Performance of ZO-Muon, LOZO, and SubZero with matched query budget and different ranks, demonstrated by fine-tuning OPT-1.3B on SST-2.



(a) OPT-1.3B on SST-2 (b) ViT-B on CIFAR-100

Figure A5: Performance of ZO-Muon with different projection matrix resampling interval v, demonstrated by (a) fine-tuning OPT-1.3B on SST-2 and (b) fine-tuning ViT-B on CIFAR-100.

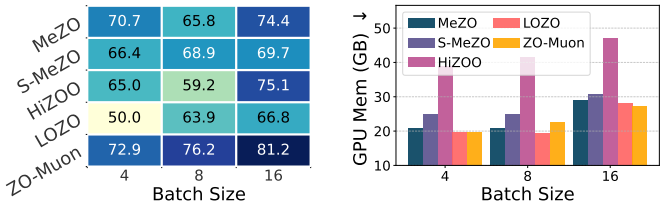
the optimizer from sufficiently exploiting gradient information within a given subspace, leading to less stable training. Our lazy resampling strategy is also aligned with previous works on both FO training [35, 68] and ZO training [18, 25]

Projection matrix sampling strategies. We present additional studies on sampling strategies for the projection matrices used in Subspace-MeZO and ZO-Muon, with results summarized in **Tab. A3**. Our default strategy, denoted as "*Random*," constructs the projection matrix $\mathbf{P} \in \mathbb{R}^{m\times r}$ via the QR decomposition of a Gaussian matrix. Alternatively, the "*Sketching*" strategy applies Gaussian sketching [42] to past descending directions to capture informative subspace directions, a technique also utilized in first-order subspace training [41, 84]. Specifically, we maintain a momentum term $\mathbf{M} \in \mathbb{R}^{m\times n}$ representing the moving average of past ZO gradients. The matrix $\mathbf{P}$ is then derived by performing QR decomposition on $\mathbf{MQ}$, where $\mathbf{Q} \in \mathbb{R}^{n\times r}$ is drawn from a Gaussian distribution. As shown in Tab. A3, Gaussian sketching does not yield notable performance gains over random sampling for either Subspace-MeZO or ZO-Muon, yet it increases GPU memory consumption, particularly for LLMs, due to the requirement of storing the momentum $\mathbf{M}$. In the ViT fine-tuning scenario, Gaussian sketching exhibits even worse accuracy, likely due to the cumulative bias introduced by Momentum as it accumulates past ZO gradient estimates rather than preciser FO gradients.

Table A3: Performance (evaluated by accuracy and F1) and GPU memory (short for "Mem") of Subspace-MeZO and ZO-Muon with different projection matrix sampling strategies.

Method	Sampling Strategy	LlaMA3-8B Fine-tuning						ViT-B Fine-tuning			
		SST-2		RTE		SQuAD		CIFAR-10		CIFAR-100	
		Acc. (%) ↑	Mem (MiB) ↓	Acc. (%) ↑	Mem (MiB) ↓	F1 (%) ↑	Mem (MiB) ↓	Acc. (%) ↑	Mem (MiB) ↓	Acc. (%) ↑	Mem (MiB) ↓
Subspace-MeZO	Random	92.3	19058	68.6	28242	84.5	32102	94.2	1634	58.2	4086
	Sketching	92.3	34814	67.5	38874	87.2	44734	81.2	1898	29.1	4088
ZO-Muon	Random	94.3	19176	81.2	27224	88.2	30718	94.9	1686	72.6	4410
	Sketching	93.8	35268	78.3	44304	88.3	45248	95.3	1950	59.8	4412

ZO-Muon does not rely on large batch sizes. We extend the evaluation of ZO-Muon with decreased batch sizes in Fig. 7 to LLMs, evaluated by fine-tuning LlaMA3-8B on RTE, shown in **Fig. A6**. The results are aligned with the ViT results: ZO-Muon outperforms the baselines by a large margin with a decreased batch size. And ZO-Muon with a batch size of 4 outperforms S-MeZO and LOZO with a batch size of 16.



(a) Fine-tuning accuracy (b) GPU memory

Figure A6: Fine-tuning accuracy and GPU memory comparison under varying batch sizes. ZO-Muon versus baselines on LlaMA3-8B fine-tuning for RTE with the same number of iterations.

J Limitations

Despite the extensive experiments and the clear performance/efficiency gain of ZO-Muon, our work has limitations that highlight future directions. First, we focus on memory-efficient fine-tuning, and extending ZO-Muon to full pretraining of large-scale models remains open. Second, exploring the application of ZO optimization in other post-training tasks, such as reinforcement learning and safety alignment, remains open.

K Ethics Statement

The research presented in this paper conforms to the NeurIPS Code of Ethics. Our research focuses on memory-efficient training of large models. This research does not involve human subjects, crowdsourcing, or the release of new models or datasets that carry a high risk of misuse. Because our contribution is primarily algorithmic and methodological, we do not foresee direct negative societal impacts arising specifically from this work.

L LLM Usage

In this paper, the sole purpose of LLMs is to assist with improving the fluency of the paper, such as refining the grammar. At no point did the language model contribute to research ideas or to the generation of original content.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix J.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Theoretical proofs are detailed in Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed setups are reported in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data used in this paper are fully open datasets. The code and instructions are included in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed experiment setup is reported in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Given the large volume of quantitative results, the paper does not explicitly report error bars for each data point.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Detailed compute resources are reported in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors confirm that in every respect, the research is not against NeurIPS code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We are proposing an algorithm for efficient training, and do not have such concerns. See Appendix K for details.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not have such safety concerns.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the open datasets and models are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our research does not concern human subjects.

    Guidelines: The paper does not involve crowdsourcing nor research with human subjects.

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components. See Appendix L for details.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.