MAPS: MOMENTUM-AWARE PERTURBATION SAMPLING FOR ZEROTH-ORDER OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Zeroth-order (ZO) optimization enables memory-efficient LLM fine-tuning through gradient estimation by forward-only function queries, but most existing methods sample queries based on random isotropic perturbations on parameters that ignore informative optimization states accumulated during training. We introduce **Momentum-Aware Perturbation Sampling (MAPS)**, a plug-and-play framework that uses momentum state to guide subsequent ZO queries. MAPS first calibrates the momentum state to the scale of isotropic perturbations and then mixes momentum-guided directions with stochastic exploration. It further balances the momentum-driven exploitation with random exploration through block-wise gating and temporal scheduling, controlling when and where the momentum-guided queries are applied. Our theoretical analysis establishes the convergence guarantee for MAPS. Since MAPS only modifies perturbation sampling without changing subsequent operations or introducing any additional function evaluation budget, it is generally applicable and extensively evaluated on various base ZO methods, foundation models and tasks. Our extensive experiments show that MAPS improves downstream performance and achieves lower training loss, demonstrating the effectiveness of reusing optimizer states in perturbation sampling.

1 INTRODUCTION

Fine-tuning large language models (LLMs) with first-order information remains memory intensive because backpropagation requires storing intermediate activations. Parameter efficient methods such as LoRA (Hu et al., 2022) reduce the number of trainable parameters by updating only low-rank adapters. QLoRA further reduces the memory overhead through quantization (Dettmers et al., 2023). Nevertheless, they still rely on backpropagation and therefore do not eliminate huge memory costs in storing activations. This limitation becomes increasingly restrictive as model size, batch size, and sequence length grow, limiting the feasibility of large models on memory-constrained hardware.

Zeroth-order (ZO) optimization provides an alternative by estimating update directions solely from function evaluations. Classical randomized ZO methods construct gradient estimates using finite differences along sampled perturbation directions (Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017). MeZO adapts this principle to LLM fine-tuning through in-place perturbations and seed-based noise regeneration, achieving an inference level memory footprint (Malladi et al., 2023). Subsequent work systematically benchmarks forward-only optimization for memory-efficient LLM adaptation (Zhang et al., 2024). However, this memory efficiency comes with an optimization challenge: in high-dimensional parameter spaces, randomly sampled perturbations can provide weak directional information, leading to slow optimization progress under a limited query budget. Since each ZO estimate requires additional forward evaluations, it is crucial to construct more informative perturbations without increasing the budget for efficient ZO fine-tuning.

Recent studies have improved ZO optimization from several complementary perspectives. Variance reduction and adaptive moment methods aim to stabilize gradient estimates and parameter updates (Liu et al., 2018; Chen et al., 2019). For LLM fine-tuning, LOZO, TeZO, and SubZero exploit structural properties of parameter matrices, temporal perturbations, or random subspaces (Chen et al., 2025b; Sun et al., 2026; Yu et al., 2025). HiZOO incorporates diagonal curvature information (Zhao et al., 2025), whereas FZOO redesigns the estimator and perturbation distribution to improve optimization speed (Dang et al., 2026). These advances improve the estimator, search space, or update rule.

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

These developments also raise the question of how existing optimizer states can be used to guide perturbation sampling.

Several studies have investigated guided perturbation sampling. Guided ES biases random search toward a subspace defined by surrogate gradients (Maheswaranathan et al., 2019). ZO-AdaMU adapts perturbations using historical perturbation momentum and uncertainty within an adaptive optimizer (Jiang et al., 2024). These studies suggest that accumulated directional information can be useful for guiding subsequent ZO queries. For optimizers with a momentum state, they already have a readily available source of such information. ZO-AdaMU develops its sampling strategy as part of a particular adaptive optimizer. We therefore seek a simple, reusable sampling mechanism that reuses the momentum already maintained by different ZO optimizers, while preserving their update rules and per-iteration function evaluation budget.

To this end, we propose **M**omentum-**A**ware **P**erturbation **S**ampling (**MAPS**), a plug-and-play sampling framework that uses the momentum maintained by the base optimizer to guide the sampling of subsequent perturbations. Our method is built upon three key components. **(1) Momentum guidance** treats this state as a directional prior, calibrates its norm to the scale of isotropic Gaussian perturbations, and mixes the two to construct subsequent queries. **(2) Block-wise gating** restricts momentum guidance to a random subset of parameter blocks to encourage parameter exploration. **(3) The temporal scheduler** uses warmup and stochastic activation to control when guidance is applied, limiting reliance on potentially unreliable historical directions. MAPS changes only the perturbation sampler while retaining the two-point finite-difference estimator (Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017), optimizer update, and function evaluation budget, so it has negligible computational overhead. We instantiate MAPS with LOZO-M and ZO-Adam, supporting low-rank and dense perturbation representations, for full-parameter and LoRA fine-tuning. We provide convergence analysis for MAPS in generic nonconvex setting and empirically evaluate MAPS against the corresponding base optimizers in various tasks and foundation models.

Our contributions are summarized as follows:

- We propose **MAPS**, a plug-and-play sampling framework that reuses existing optimizer states by combining calibrated momentum guidance with isotropic exploration. It further applies selective guidance across blocks and iterations without additional memory overhead.
- We establish a nonconvex convergence guarantee for MAPS-LOZO-M under the stated assumptions, with LOZO-M and LOZO recovered as special cases.
- We instantiate MAPS with LOZO-M and ZO-Adam, and evaluate it across different model architectures and scales, under both full-parameter and LoRA fine-tuning settings. The results demonstrate improved downstream performance and favorable empirical convergence.

2 Related work

Memory-Efficient Fine-Tuning. Parameter-efficient fine-tuning methods reduce the number of trainable parameters and the optimizer states by updating only a small subset of parameters or by using a compact parameterization of the model. Representative approaches introduce adapter modules (Houlsby et al., 2019), optimize continuous prefixes or prompts (Lester et al., 2021; Li & Liang, 2021), update only bias terms (Zaken et al., 2022), or parameterize weight updates with trainable low-rank matrices (Hu et al., 2022). Although these methods reduce memory consumption, they still rely on backpropagation and thus require storing activations, which account for the major memory overhead. This motivates zeroth-order optimization, which avoids backpropagation computation.

Zeroth-Order Optimization. Classical ZO methods, such as SPSA, estimate gradients from function evaluations under random perturbations (Spall, 1987; 1992). MeZO extends this paradigm to memory-efficient LLM fine-tuning via in-place perturbation and noise regeneration (Malladi et al., 2023). Subsequent studies improve gradient estimation from different aspects, including variance reduction in MeZO-SVRG (Gautam et al., 2024), multiple queries in MUZO (Peng et al., 2025) and reduced effective optimization dimension in later work (Chen et al., 2024; Liu et al., 2025; Guo et al., 2025). More recent approaches further incorporate structural priors into ZO optimization, including curvature information and low-rank structures in matrix parameter updates (Chen et al., 2025b; Sun et al., 2026; Yu et al., 2025; Zhao et al., 2025; Seung et al., 2026).

ZO methods are usually computationally expensive, because of gradient estimation and slower convergence. To address this issue, LoHO and Addax combine first order and zeroth order updates for faster convergence, while QuZO reduces computation of gradient estimation through low precision forward evaluation (Chen et al., 2025a; Li et al., 2025; Zhou et al., 2025). FZOO further improves computational efficiency through batched estimation and perturbations suited to efficient tensor computation (Dang et al., 2026).

Guided Perturbation Sampling. Earlier ZO methods query parameter neighborhood with isotropic random Gaussian perturbations to estimate gradients in that direction. Besides isotropic perturbations, several methods bias queries toward informative directions. Guided ES biases random search toward a subspace defined by surrogate gradients (Maheswaranathan et al., 2019), while ASEBO learns an active subspace from previous evolution strategy estimates and allocates sampling between this subspace and its orthogonal complement (Choromanski et al., 2019). For LLMs, ZO-AdaMU adapts perturbation queries by momentum and uncertainty (Jiang et al., 2024). More recent methods explore targeted or learned sampling strategies: ZOPrO exploits policy and reward update trajectories to construct targeted perturbations for preference optimization (Galatolo et al., 2025), ZO Fine-tuner learns adaptive nonuniform perturbations (Zhang et al., 2026), and CurvZO constructs parameter sampling distributions from online curvature information (Wang et al., 2026). These methods show that biased queries can help. ZO-AdaMU already uses momentum to shape the next perturbation, but it does so inside its specialized adaptive optimizer. MAPS instead leverages the momentum already stored by a given base optimizer and changes only the sampler. It can be seamlessly integrated with diverse base optimizers.

3 PRELIMINARIES

Let $F(\boldsymbol{\theta};\xi)$ denote the stochastic loss objective on a minibatch ξ, where $\boldsymbol{\theta} \in \mathbb{R}^d$ represents the trainable parameters, and let $f(\boldsymbol{\theta}) = \mathbb{E}_\xi[F(\boldsymbol{\theta};\xi)]$ be the expected objective. ZO optimization estimates the gradients through function evaluations without explicitly computing them. Given a sampled perturbation direction $\boldsymbol{z}$ and a perturbation radius $\mu > 0$, the two-point ZO estimator (Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017) evaluates the same minibatch at two symmetric perturbations:

$$c = \frac{F(\boldsymbol{\theta} + \mu\boldsymbol{z};\xi) - F(\boldsymbol{\theta} - \mu\boldsymbol{z};\xi)}{2\mu}, \qquad \widehat{\boldsymbol{g}} = c\boldsymbol{z}, \tag{1}$$

where $\widehat{\boldsymbol{g}}$ is the gradient estimate. Equation (1) demonstrates the gradient estimation with one sampled perturbation direction (Malladi et al., 2023). When Q directions $\{\boldsymbol{z}_q\}_{q=1}^Q$ are sampled, we calculate the magnitude estimation $\{c_q\}_{q=1}^Q$ similar to (1) and estimate the gradient by the average, i.e., $\widehat{\boldsymbol{g}} = \frac{1}{Q}\sum_{q=1}^Q c_q\boldsymbol{z}_q$.

A common ZO baseline independently samples each direction from an isotropic Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$. The obtained stochastic gradient estimate $\widehat{\boldsymbol{g}}$ can be incorporated into optimizers with momentum states and preconditioners for parameter updates. For example, ZO-Adam uses zeroth-order gradient estimation $\widehat{\boldsymbol{g}}$ in the update scheme of Adam (Kingma & Ba, 2015) as follows:

$$\begin{cases} \boldsymbol{m}_t \leftarrow \beta_1\boldsymbol{m}_{t-1} + (1-\beta_1)\widehat{\boldsymbol{g}}_t, \\ \boldsymbol{v}_t \leftarrow \beta_2\boldsymbol{v}_{t-1} + (1-\beta_2)\widehat{\boldsymbol{g}}_t^2, \end{cases} \qquad \boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta\frac{\boldsymbol{m}_t/(1-\beta_1^t)}{\sqrt{\boldsymbol{v}_t/(1-\beta_2^t)} + \epsilon}, \tag{2}$$

where $\widehat{\boldsymbol{g}}_t$, $\boldsymbol{m}_t$ and $\boldsymbol{v}_t$ represent the gradient estimation, the first-order and the second-order momentum in iteration t, respectively. In addition, β_1, β_2 are momentum coefficients, η is the learning rate, and ϵ is a small constant to avoid zero division.

Equation (1) treats model parameters as a vector and does not exploit their structural information. By contrast, recent ZO methods for LLM fine-tuning exploit low-rank subspaces when optimizing matrix parameters. For example, LOZO estimates gradients for a matrix parameter $\mathbf{X} \in \mathbb{R}^{m\times n}$ by:

$$c = \frac{F(\mathbf{X} + \mu\mathbf{U}\mathbf{V}^T;\xi) - F(\mathbf{X} - \mu\mathbf{U}\mathbf{V}^T;\xi)}{2\mu}, \qquad \widehat{\boldsymbol{g}} = c\cdot\frac{1}{r}\mathbf{U}\mathbf{V}^T, \tag{3}$$

where $\mathbf{U} \in \mathbb{R}^{m\times r}$, $\mathbf{V} \in \mathbb{R}^{n\times r}$ are random Gaussian matrices with a lower rank r. In practice, the right factor $\mathbf{V}$ is fixed within an interval of ν steps and sampled only periodically. LOZO-M further

introduces momentum for the left factor $\mathbf{U}$, it updates parameters by:

$$\mathbf{N}_t \leftarrow \beta \mathbf{N}_{t-1} + (1-\beta) c_t \mathbf{U}_t, \qquad \mathbf{X}_{t+1} \leftarrow \mathbf{X}_t - \eta \cdot \mathbf{N}_t \mathbf{V}_t^T / r, \tag{4}$$

where $\mathbf{N}_t$ is the momentum in iteration t, c_t is the magnitude calculated in (3) in iteration t, and β is the momentum coefficient. When the right factor $\mathbf{V}_t$ is resampled after a fixed period, LOZO-M projects the momentum state $\mathbf{N}_t$ onto the new low-rank subspace by $\mathbf{V}_{t+1}$ to preserve historical momentum information, i.e., $\mathbf{N}_t \leftarrow \arg\min_{\mathbf{N}} \|\mathbf{N}_t \mathbf{V}_t^T - \mathbf{N}\mathbf{V}_{t+1}^T\|$.

Both ZO-Adam in (2) and LOZO-M in (4) maintain optimization states, such as momentum, that summarize historical gradient estimates. However, these ZO methods still rely on isotropic Gaussian perturbations for gradient estimation. In this work, we explore how optimization states can be leveraged as priors for perturbations in ZO gradient estimation to improve the performance of the overall optimization pipeline.

4 METHODOLOGY

We propose **Momentum-Aware Perturbation Sampling (MAPS)**, a plug-and-play framework for ZO optimization that uses directional information from an optimizer's momentum state to guide subsequent queries. MAPS only modifies the perturbation sampler while keeping the optimizer's update scheme and function evaluation budget unchanged.

4.1 MOMENTUM GUIDANCE FOR PERTURBATION SAMPLING

We consider first-order Taylor expansion on the two-point estimator in Equation (1) and obtain $c \approx \boldsymbol{z}^\top \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}; \xi)$, so that $\widehat{\boldsymbol{g}} \approx \left(\boldsymbol{z}^\top \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}; \xi)\right)\boldsymbol{z} = \|\boldsymbol{z}\|^2 \left(\frac{\boldsymbol{z}^\top}{\|\boldsymbol{z}\|} \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}; \xi)\right) \frac{\boldsymbol{z}}{\|\boldsymbol{z}\|}$. When $\|\boldsymbol{z}\|$ has a fixed expectation, the estimate is proportional to the projection of the true stochastic gradient $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}; \xi)$ onto the direction of the sampled perturbation $\boldsymbol{z}$. Therefore, $\boldsymbol{z}$ with a direction more aligned with $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}; \xi)$ recovers more of the true gradient, whereas samples with a nearly orthogonal direction will make the estimation $\widehat{\boldsymbol{g}} \approx \mathbf{0}$ and lead to slower convergence. Intuitively, **the momentum state of the optimizer**, which averages past gradient estimates, is utilised as **a prior for the gradient** and is incorporated into perturbation sampling. We can adopt a similar strategy for the estimator with structural information in Equation (3), where the right factor stays fixed within a lazy interval and we incorporate the momentum into the sampling of the left factor.

MAPS constructs blockwise perturbations, so we consider a block with a d-dimensional parameter without the loss of generality. Let $\boldsymbol{R} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ denote a fresh random exploration component, and let $\boldsymbol{P} \in \mathbb{R}^d$ denote a directional momentum state obtained from the optimizer before the current function evaluations. The perturbation sample $\boldsymbol{S}$ is then a combination of $\boldsymbol{R}$ and $\boldsymbol{P}$.

MAPS applies to any ZO method with momentum, so $(\boldsymbol{R}, \boldsymbol{P}, \boldsymbol{S})$ is an abstract tuple in the generic formulation and may denote different concrete quantities for different optimizers. For example, when we apply MAPS to ZO-Adam (**MAPS-ZO-ADAM**), $\boldsymbol{S}$ is the perturbation $\boldsymbol{z}$ in Equation (1), $\boldsymbol{P}$ is the momentum $\boldsymbol{m}$ in Equation (2); when we apply MAPS to LOZO-M (**MAPS-LOZO-M**), $\boldsymbol{S}$ is the left factor $\boldsymbol{U}$ in Equation (3), $\boldsymbol{P}$ is the momentum $\boldsymbol{N}$ in Equation (4); in both cases, $\boldsymbol{R}$ is a Gaussian variable of the same shape as $\boldsymbol{P}$ and $\boldsymbol{S}$.

The magnitude of the momentum state $\boldsymbol{P}$ is generally not calibrated to that of the random perturbation $\boldsymbol{R}$. Directly mixing them would therefore make the effective strength of momentum guidance depend on the magnitude of the momentum. Considering $\mathbb{E}\|\boldsymbol{R}\|_F^2 = d$, we normalize the momentum $\boldsymbol{P}$ to match the magnitude of $\boldsymbol{R}$ and then mix them to obtain the perturbation sample $\boldsymbol{S}$ as follows:

$$\widetilde{\boldsymbol{P}} = \sqrt{d} \frac{\boldsymbol{P}}{\max\{\|\boldsymbol{P}\|_F, \epsilon\}}, \quad \boldsymbol{S} = \sqrt{1-\lambda}\boldsymbol{R} + \sqrt{\lambda}\widetilde{\boldsymbol{P}}, \;\; \lambda \in [0,1], \tag{5}$$

where $\epsilon > 0$ is a small constant used to avoid division by zero, and λ controls the relative contributions of random exploration and momentum exploitation. When $\|\boldsymbol{P}\|_F \le \epsilon$, i.e., when no valid momentum is available, we set $\lambda = 0$ to fall back to a random sampler. A larger λ induces a stronger directional bias toward the momentum. Moreover, the normalized momentum $\widetilde{\boldsymbol{P}}$ is used only to compute the perturbation sample $\boldsymbol{S}$, whereas the original momentum $\boldsymbol{P}$ is still used for the its own update (e.g., Equations (2) and (4)), because the gradient estimate $\widehat{\boldsymbol{g}}$ is not normalized.

Since $\boldsymbol{R}$ is a Gaussian random variable, we have $\mathbb{E}[\boldsymbol{R}] = \mathbf{0}$ and $\mathbb{E}[\|\boldsymbol{R}\|_F^2] = d$. Unlike the zero-mean isotropic sampler, MAPS introduces an explicit directional preference toward the accumulated optimizer state. Considering $\boldsymbol{R}$ does not depend on $\boldsymbol{P}$, we have: $\mathbb{E}[\boldsymbol{S}] = \sqrt{\lambda}\widetilde{\boldsymbol{P}},\ \mathbb{E}[\|\boldsymbol{S}\|_F^2] = (1-\lambda)\mathbb{E}\|\boldsymbol{R}\|_F^2 + \lambda\|\widetilde{\boldsymbol{P}}\|_F^2 = d$. Therefore, MAPS changes the directional distribution of ZO queries while preserving their expected squared norm.

4.2 Selective activation of momentum guidance

Section 4.1 treats the momentum state as a reliable estimate of the gradient. This does not always hold in ZO optimization, because the momentum is accumulated from biased ZO gradient estimates. Guiding the next perturbation with such a state creates a positive feedback mechanism: once the estimate drifts away from the true stochastic gradient, the sampler concentrates its queries around the wrong direction, and the momentum accumulated from those queries drifts further. To address this issue, we adopt mechanisms similar to Dropout (Srivastava et al., 2014). We randomly disable the momentum guidance for a fraction of the parameter blocks and for a fraction of the steps. Specifically, **a block-wise gate** decides **where the guidance is applied**, and **a temporal scheduler** decides **when it is active**. Both mechanisms are designed to preserve the exploration capabilities of the algorithm while still exploiting historical directional signal.

Block-wise Gating. Let $\mathcal{B} = \{1, \dots, B\}$ index the parameter blocks of the model. At each iteration t, MAPS applies momentum guidance only to a subset of eligible parameter blocks. We define $\mathcal{S}_t \subseteq \mathcal{B}$ as the selected subset and ensure $|\mathcal{S}_t| = \mathrm{round}(pB)$ where $p \in [0,1]$ denotes the selection ratio. Furthermore, $\mathcal{S}_t$ is sampled uniformly from the eligible blocks and may be retained for several optimization steps before being resampled. For a selected block $\ell \in \mathcal{S}_t$, MAPS applies the momentum guided sampling rule in Equation (5). For an unselected block $\ell \notin \mathcal{S}_t$, we set $\boldsymbol{S} = \boldsymbol{R}$.

Temporal Scheduler. The momentum state is less informative during the early stage of optimization, so MAPS uses the random sampler, i.e., $\boldsymbol{S} = \boldsymbol{R}$, during an initial warmup period of $W = \lfloor \rho T \rfloor$, where T is the total number of optimization steps and $\rho \in [0,1]$ is the warmup ratio. Let λ_t denote the value of the mixing coefficient λ in Equation (5) in iteration t, then we set λ_t as follows: $\lambda_t = \mathbf{1}(t > W) \cdot \lambda_{\max} b_t,\ b_t \sim \mathrm{Bernoulli}(q)$, where $\mathbf{1}$ is the indicator function and $\lambda_{\max}$ indicates the largest possible value for λ. Therefore, after warmup, MAPS uses the momentum guided perturbation with probability q and uses the random perturbation with probability $1-q$. This stochastic activation mechanism preserves random exploration throughout optimization while selectively exploiting the accumulated directional information.

The block selection and temporal scheduler serve complementary roles. The block selection determines where historical directional information is used, whereas the temporal scheduler controls when it is introduced. Together, they allow MAPS to exploit informative momentum states while retaining the random exploration of the sampler.

4.3 Plug-and-Play Instantiation of MAPS

MAPS applies to any ZO optimizer that maintains a momentum state. Algorithm 1 presents the procedure of MAPS for a generic base optimizer $\mathcal{A}$, where $\Pi(\cdot)$ maps a perturbation representation to a perturbation in parameter space and d_ℓ is the dimension of the momentum state for block ℓ. MAPS only modifies the perturbation samples $\boldsymbol{S}$, so the two-point estimator, the number of function evaluations per step, the momentum update, and the parameter update remain those of $\mathcal{A}$. An instantiation is therefore specified by the tuple $(\boldsymbol{R}, \boldsymbol{P}, \boldsymbol{S})$ alone, as the two examples below show.

For ZO-Adam with its update rules in Equations (1), (2), MAPS operates in the dense perturbation space, with $\boldsymbol{S} = \boldsymbol{z}$, $\boldsymbol{P} = \boldsymbol{m}$ and $\Pi(\boldsymbol{S}) = \boldsymbol{S}$, then the first moment, the second moment, and the parameter update rules follow ZO-Adam. The same construction applies when we use LoRA parameterization (Hu et al., 2022) for LLMs, with the eligible blocks restricted to the LoRA parameters. Algorithm 2 in Appendix A gives the full pseudocode for MAPS-ZO-ADAM.

For LOZO-M with its update rules in Equations (3), (4), MAPS operates in the left factor space of the low-rank perturbation, with $\boldsymbol{S} = \boldsymbol{U}$, $\boldsymbol{P} = \mathbf{N}$, and $\Pi(\boldsymbol{S}) = \boldsymbol{S}\boldsymbol{V}^\top$. Then interval updates in $\boldsymbol{V}$, the projection of $\boldsymbol{N}$ onto the new subspace, parameter update rules remain unchanged. Algorithm 3 in Appendix A gives the full pseudocode for MAPS-LOZO-M.

Algorithm 1 MAPS (Generic Form)

Require: Base ZO optimizer $\mathcal{A}$ with perturbation representation $\boldsymbol{R}$ and momentum state $\boldsymbol{P}$, parameter blocks $\{\boldsymbol{\theta}_\ell\}_{\ell=1}^{B}$, steps T, perturbation radius μ, guidance strength $\lambda_{\max}$, warmup ratio ρ, activation probability q, block ratio p, threshold ϵ.

Initialize the states of $\mathcal{A}$ and set $W \leftarrow \lfloor \rho T \rfloor$
for $t = 1, \dots, T$ **do**
  $\lambda_t = \mathbf{1}(t > W) \cdot \lambda_{\max} b_t$ with $b_t \sim \text{Bernoulli}(q)$ ▷ **Temporal Scheduler**
  Sample $\mathcal{S}_t \subseteq \mathcal{B}$ with $|\mathcal{S}_t| = \text{round}(pB)$ ▷ **Block-wise Gating**
  for $\ell = 1, \dots, B$ **do**
    Sample the perturbation representation $\boldsymbol{R}_{t,\ell}$ as in $\mathcal{A}$
    if $\ell \in \mathcal{S}_t$, $\lambda_t > 0$ and $\|\boldsymbol{P}_{t-1,\ell}\|_F > \epsilon$ **then**
      $\boldsymbol{S}_{t,\ell} \leftarrow \sqrt{1-\lambda_t}\, \boldsymbol{R}_{t,\ell} + \sqrt{\lambda_t d_\ell}\, \dfrac{\boldsymbol{P}_{t-1,\ell}}{\max\{\|\boldsymbol{P}_{t-1,\ell}\|_F, \epsilon\}}$ ▷ **Momentum-Guided Sampling**
    else $\boldsymbol{S}_{t,\ell} \leftarrow \boldsymbol{R}_{t,\ell}$
  $F_\pm \leftarrow F(\boldsymbol{\theta}_t \pm \mu\, \Pi(\boldsymbol{S}_t); \xi_t)$; $\quad c_t \leftarrow (F_+ - F_-)/(2\mu)$
  Update the states and parameters of $\mathcal{A}$ from c_t and $\boldsymbol{S}_t$

4.4 Theoretical analysis

Building on the aforementioned definitions, we establish the convergence guarantee of MAPS-LOZO-M under the following standard assumptions in stochastic optimization.

Assumption 4.1. *The loss objective $F(\boldsymbol{X}; \xi)$ is differentiable for any parameter $\mathbf{X}$ and data ξ. It is Lipschitz smooth with a constant $L > 0$: $\|\nabla F(\boldsymbol{X}; \xi) - \nabla F(\boldsymbol{Y}; \xi)\| \le L\|\boldsymbol{X} - \boldsymbol{Y}\|$. Moreover, the stochastic gradient is unbiased and has bounded variance: $\mathbb{E}_\xi[\nabla F(\boldsymbol{X}; \xi)] = \nabla f(\boldsymbol{X})$ and $\mathbb{E}_\xi\|\nabla F(\boldsymbol{X}; \xi) - \nabla f(\boldsymbol{X})\|^2 \le \sigma^2$, where $f(\boldsymbol{X}) := \mathbb{E}_\xi[F(\boldsymbol{X}; \xi)]$. In addition, we assume that f is lower bounded by $f_\star > -\infty$ and define $\Delta_0 := f(\boldsymbol{X}_0) - f_\star$.*

Assumption 4.2. *[Kozak et al. (2023); Chen et al. (2025b)] For a matrix block, the right factor $\boldsymbol{V}$ satisfies $\boldsymbol{V}^\top \boldsymbol{V} = n\boldsymbol{I}_r$ and $\mathbb{E}[\boldsymbol{V}\boldsymbol{V}^\top] = r\boldsymbol{I}_n$.*

Considering that the block subset is sampled independently, we define $\widetilde{\lambda} := pq\lambda_{\max}$. During the warmup stage or when the momentum state is unavailable, MAPS reduces to the original random sampler. Hence, the actual expected momentum guidance strength is no larger than $\widetilde{\lambda}$. We are now ready to establish the convergence guarantee as follows:

Theorem 4.3 (Simplified Convergence of MAPS-LOZO-M)**.** *Under Assumptions 4.1 and 4.2, consider a single matrix block $\boldsymbol{X} \in \mathbb{R}^{m \times n}$. Let $T = K\nu$, $\beta \in [0, 1)$, $d = mr$ and $\widetilde{\lambda} < 1$. With suitable choices of η and μ, MAPS-LOZO-M satisfies the following convergence bound after T iterations:*

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\|\nabla f(\boldsymbol{X}_{k\nu})\|_F^2 \le \frac{C_1\sqrt{(1+\beta)(1+d\widetilde{\lambda})}}{(1-\beta)^{3/2}(1-\widetilde{\lambda})}\sqrt{\frac{\Delta_0 L m n^2 \sigma^2}{rT}} + \frac{C_2(1+d\widetilde{\lambda})\Delta_0 L m n \nu}{(1-\beta)^3(1-\widetilde{\lambda})T}. \quad (6)$$

where $C_1, C_2 > 0$ are positive constants.

The parameter choices conditions, the exact value of C_1, C_2 and proof are provided in Appendix B. The bound retains the $\mathcal{O}(T^{-1/2})$ convergence rate of LOZO-M, with the effect of momentum guidance captured by $\widetilde{\lambda}$ in the multiplicative constants. When $\widetilde{\lambda} = 0$, the bound in (6) reduces to that for LOZO-M. When $\widetilde{\lambda} = 0$, $\beta = 0$, the bound in (6) reduces to that for LOZO (Chen et al., 2025b).

5 Experiments

Experimental Setup. We fine-tune the decoder-only OPT models (1.3B, 13B and 30B) (Zhang et al., 2022) on the 10 downstream tasks used in prior work on ZO methods, including classification tasks (SST-2, RTE, CB, BoolQ, WiC, MultiRC), multiple choice tasks (COPA, ReCoRD), and question-answering tasks (SQuAD, DROP). We report F1 for question-answering tasks and accuracy for the rest. MAPS modifies only the perturbation sampler of an existing optimizer, so we compare

MAPS-LOZO-M with LOZO-M and MAPS-ZO-ADAM with ZO-Adam. Each pair shares the same update rules and function evaluation budget, so MAPS differs from the base optimizer only in its momentum-guided perturbation sampling. We follow the experimental settings commonly used in ZO fine-tuning literature (Malladi et al., 2023; Chen et al., 2025b). Besides the maximum mixing coefficient $\lambda_{\max}$ and the block selection ratio p, we adopt the same settings as in baselines for MAPS, including training budget, batch size, perturbation radius, learning rate, and optimizer configurations. The detailed settings are deferred to Appendices D.1 and D.3.

5.1 Main Results

Overall Performance. Table 1 reports OPT-13B results on the different tasks, where we fine-tune the model for 20000 iterations using LOZO-M, ZO-Adam, ZO-Adam (LoRA) and their MAPS variants. The table also lists zero-shot inference, in-context learning (ICL) (Brown et al., 2020), full-parameter fine-tuning (FT) (Kingma & Ba, 2015; Hu et al., 2022), MeZO (Malladi et al., 2023), and ZO-AdaMU (Jiang et al., 2024) as reference methods. Across LOZO-M, ZO-Adam, and ZO-Adam (LoRA), MAPS achieves an average improvement of 1.5% over the corresponding base optimizers. Improvements are observed on most tasks. These results demonstrate the effectiveness of MAPS across both low-rank and dense perturbation spaces, as well as full-parameter and LoRA fine-tuning.

Table 1: The performance of various methods and tasks on OPT-13B. Bold numbers indicate the better result between each MAPS instance and its corresponding base optimizer. Results are reported in percentage. Baseline results taken from prior work are indicated by: †: Fang et al. (2026); ‡: Jiang et al. (2024); *: Chen et al. (2025b) (subset of tasks).

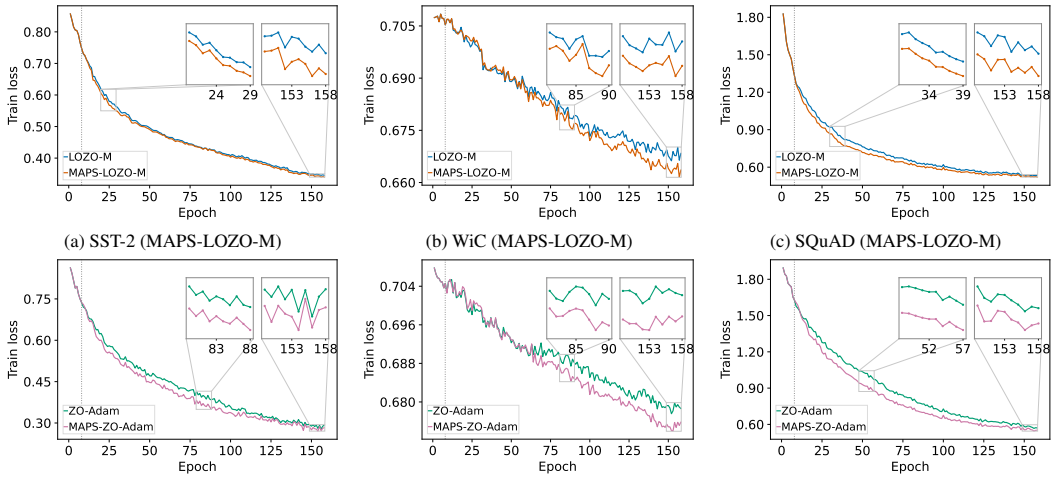
Method	Classification						Multiple choice		Question answering		Average
	SST-2	RTE	CB	BoolQ	WiC	MultiRC	COPA	ReCoRD	SQuAD	DROP	
Zero-shot†	58.8	59.6	46.4	59.0	55.0	46.9	80.0	81.0	46.2	14.6	54.8
ICL†	87.0	62.1	57.1	66.9	50.5	53.1	87.0	82.3	75.9	29.5	65.1
FT†	92.0	70.8	83.9	77.1	70.1	71.1	79.0	74.1	84.9	31.3	73.4
FT (LoRA)†	92.4	62.1	60.7	66.9	58.2	58.0	84.0	81.1	81.3	29.4	67.4
MeZO†	91.3	68.2	66.1	68.1	59.4	59.4	88.0	81.3	81.8	31.3	69.5
MeZO (LoRA)†	89.6	67.9	67.8	73.5	60.2	61.3	84.1	81.5	82.1	31.3	69.9
ZO-AdaMU‡	92.1	72.9	67.9	73.0	60.7	63.0	89.0	83.0	82.4	32.0	71.6
ZO-AdaMU (LoRA)‡	88.0	72.0	71.6	72.6	56.4	58.9	88.0	83.2	76.8	32.4	70.0
LOZO-M	92.5*	73.6*	69.6*	70.3	59.2	59.3	**89.0**	81.5	83.3*	29.1	70.7
MAPS-LOZO-M	**93.3**	**74.0**	**71.4**	**71.2**	**61.6**	**64.1**	**89.0**	**82.0**	**84.1**	**31.1**	**72.2**
ZO-Adam	92.9	**75.5**	66.1	72.9	63.0	60.9	**88.0**	81.0	84.4	**28.9**	71.3
MAPS-ZO-ADAM	**93.2**	74.7	**69.6**	**73.6**	**63.8**	**65.7**	**88.0**	**81.2**	**85.1**	28.5	**72.4**
ZO-Adam (LoRA)	93.0	71.5	69.6	73.8	58.6	58.4	85.0	81.6	85.7	**32.1**	70.9
MAPS-ZO-ADAM (LoRA)	**93.6**	**74.4**	**73.2**	**76.2**	**61.3**	**60.3**	**88.0**	**82.1**	**86.5**	31.5	**72.7**

Scaling to OPT-30B. We further evaluate MAPS-LOZO-M on OPT-30B using five tasks from the main benchmark. Table 2 compares it with LOZO-M under the same query budget. MAPS-LOZO-M raises the average from 73.36% to 75.07%. Peak GPU memory is identical for the two methods. These results show that the sampling rule remains effective when the model scales to 30B.

Table 2: Comparison between LOZO-M and MAPS-LOZO-M on OPT-30B and various tasks. Peak GPU memory is the same for both methods and ranges from 56.5 to 61.4 GB. Bold numbers mark the better result. Results are reported in percentage.

Method	SST-2	CB	BoolQ	WiC	SQuAD	Avg.
LOZO-M	91.97	**69.64**	70.00	51.10	84.10	73.36
MAPS-LOZO-M	**92.09**	**69.64**	**70.60**	**57.84**	**85.20**	**75.07**

Convergence Analysis. To examine the convergence of MAPS, we compare MAPS-LOZO-M with LOZO-M and MAPS-ZO-ADAM with ZO-Adam on SST-2, WiC, and SQuAD using OPT-1.3B. All methods are trained for 10000 steps under the same settings and function evaluation budget. Figure 1 presents the corresponding training loss curves. After warmup, MAPS generally exhibits lower training loss and attains a lower final epoch loss in all six comparisons. These results indicate that momentum-aware perturbation sampling can produce a more favorable convergence trajectory under the same function evaluation budget.



(a) SST-2 (MAPS-LOZO-M) (b) WiC (MAPS-LOZO-M) (c) SQuAD (MAPS-LOZO-M)

(d) SST-2 (MAPS-ZO-ADAM) (e) WiC (MAPS-ZO-ADAM) (f) SQuAD (MAPS-ZO-ADAM)

Figure 1: Training loss curves of different tasks and methods on OPT-1.3B.

Table 3: Comparison between MAPS-LOZO-M and LOZO-M on different models. Results are reported in percentage. Bold numbers mark the better result.

	OPT-1.3B				Qwen2.5-1.5B				Llama-3.2-1B			
Method	SST-2	WiC	CB	SQuAD	SST-2	WiC	CB	SQuAD	SST-2	WiC	CB	SQuAD
LOZO-M	87.61	60.50	71.43	74.49	90.02	56.90	**87.50**	80.58	88.99	53.76	**67.86**	75.81
MAPS-LOZO-M	**90.14**	**60.82**	**75.00**	**76.08**	**90.83**	**61.13**	**87.50**	**82.86**	**90.71**	**55.80**	**67.86**	**78.84**

Table 4: Averaged performance on OPT-1.3B across 5 random seeds. Each pair uses the same splits and base optimizer hyperparameters. The better results are in bold.

Task	LOZO-M	MAPS-LOZO-M	Δ Mean	Δ Std.
SST-2	87.02 ± 1.35	**87.73** ± 1.44	$+0.71$	$+0.09$
CB	69.29 ± 1.96	**70.36** ± 0.98	$+1.07$	-0.98
SQuAD	72.42 ± 1.80	**73.48** ± 1.67	$+1.06$	-0.13

5.2 Generality, Stability, and Overhead

Generalization across Different Model Families. We compare MAPS-LOZO-M with LOZO-M on OPT-1.3B (Zhang et al., 2022), Qwen2.5-1.5B (Bai et al., 2023; Qwen et al., 2025) and Llama-3.2-1B (Grattafiori et al., 2024). All methods are trained for 10000 steps; detailed settings are provided in Appendix D.2. Table 3 shows that MAPS-LOZO-M matches or exceeds LOZO-M across these models, demonstrating that the proposed sampling strategy generalizes beyond the OPT architecture.

Robustness to Data Sampling. To evaluate the robustness of MAPS to data sampling, we resample the training data and fine-tune OPT-1.3B with five random seeds. As shown in Table 4, MAPS-LOZO-M raises the mean on three tasks with comparable or lower standard deviations. These results indicate that MAPS provides performance gains while maintaining stability.

Computational Overhead. MAPS reuses the momentum already maintained by the base optimizer, so the peak GPU memory does not increase. Appendix C.1 reports the relative per-iteration wall-clock overhead, peak GPU memory and absolute per-iteration runtime on OPT-13B on the same GPU. The per-iteration time increases by 1.4%–6.3%.

5.3 Ablation and Sensitivity Studies

Ablation Study on Components. We evaluate the components of MAPS-LOZO-M on WiC with OPT-1.3B by removing warmup, stochastic activation, momentum normalization, square-root weighting, and block-wise gating one at a time, while keeping all other training settings unchanged. Removing stochastic activation sets $b_t = 1$ after warmup, whereas removing square-root weighting replaces $\sqrt{1-\lambda}$ and $\sqrt{\lambda}$ with $1-\lambda$ and λ, respectively.

As shown in Table 5, the full MAPS-LOZO-M configuration achieves the highest accuracy and lowest training loss among the evaluated variants. Removing warmup reduces accuracy by 0.94%, consistent

Table 5: Ablation study on WiC with OPT-1.3B ($r = 4$, $\eta = 10^{-7}$, $\mu = 10^{-4}$, $T = 10K$). MAPS-LOZO-M uses $p = 0.5$ and $\lambda_{\max} = 0.3$. Δ Acc. is the change relative to MAPS-LOZO-M.

Variant	Acc. (%) ↑	Loss ↓	Δ Acc.
LOZO-M	57.52	0.66837	–
MAPS-LOZO-M	**60.50**	**0.66523**	–
w/o warmup	59.56	0.66537	−0.94
w/o stochastic activation	59.25	0.66908	−1.25
w/o momentum norm.	59.25	0.66689	−1.25
w/o square-root weighting	58.78	0.66823	−1.72
w/o block-wise gating	59.87	0.66735	−0.63

Table 6: MAPS-LOZO-M with $(\lambda_{\max}, p)$ selected on OPT-350M and evaluated on OPT-13B. † reuses the 350M pair; ‡ searches $(\lambda_{\max}, p)$ on 13B.

Method	SST-2	CB	SQuAD	Avg.
MAPS-LOZO-M (350M)	84.6	69.6	51.7	68.63
LOZO-M (13B)	92.5	69.6	83.3	81.80
MAPS-LOZO-M (13B)†	93.2	71.4	83.7	82.77
MAPS-LOZO-M (13B)‡	93.3	71.4	84.1	82.93

with the benefit of allowing momentum to accumulate before using it to guide sampling. Disabling stochastic activation and block-wise gating reduces accuracy by 1.25% and 0.63%, respectively. These results suggest that retaining unguided sampling across iterations and parameter blocks helps balance the use of historical directions with random exploration. Removing momentum normalization reduces accuracy by 1.25%, supporting the use of norm calibration to control the relative scale of momentum guidance and Gaussian exploration. Replacing square-root weighting with linear weighting produces the largest accuracy decrease of 1.72%. This substitution changes both the expected squared perturbation norm and the relative guidance strength, so the comparison reflects their combined effect. Overall, these results provide empirical support for calibrated perturbation mixing together with selective momentum guidance.

Search Cost of $\lambda_{\max}$ and p. MAPS introduces two additional hyper-parameters: $\lambda_{\max}$ and p, so it is crucial for practitioners to determine the optimal choice for them at an affordable tuning cost. In this context, we can tune these hyper-parameters on a smaller model and transfer the selected values to a larger one. We fix settings shared with the baselines and use grid search for $\lambda_{\max} \in \{0.1, 0.3, 0.5\}$ against $p \in \{0.5, 0.7, 0.9\}$ on OPT-350M on SST-2, CB, and SQuAD. For each task we keep the pair with the best OPT-350M performance and apply it to OPT-13B without further tuning. As Table 6 shows, the transferred pairs improve the average over LOZO-M by 0.97 points and come within 0.2 points of the pairs searched on OPT-13B. Under a limited compute budget, $\lambda_{\max}$ and p can therefore be selected efficiently on a small proxy model rather than at the target scale.

Block Ratio and Guidance Strength. Table 8 (Appendix C.2) evaluates the sensitivity of MAPS-LOZO-M to the block ratio p and guidance strength $\lambda_{\max}$ under the same setting as the ablation study. When varying one hyperparameter, the other is kept fixed. For the block ratio, $p = 0.5$ achieves the best performance, with the highest accuracy and lowest loss. Performance does not improve monotonically as more blocks receive momentum guidance, indicating that applying guidance to a moderate subset of parameter blocks provides a better balance than sparse or global activation. For the guidance strength, $\lambda_{\max} = 0.3$ yields the highest accuracy, while $\lambda_{\max} = 0.5$ gives the lowest loss. Increasing $\lambda_{\max}$ further does not consistently improve performance, which suggests that overly strong momentum guidance may suppress the random exploration.

6 Conclusion

We introduced **M**omentum-**A**ware **P**erturbation **S**ampling (**MAPS**), a plug-and-play framework that leverages optimizer states to guide ZO perturbation sampling. MAPS combines norm-calibrated momentum guidance with stochastic exploration. And it controls where and when momentum guidance is applied through block-wise gating and temporal scheduling, respectively. It does not change the subsequent operations of base optimizers or require additional function evaluations, demonstrating its applicability to multiple base ZO optimizers. We established convergence for MAPS. Experiments show better performance and lower training loss across low-rank and dense perturbation spaces, highlighting the effectiveness and practicality of momentum guidance.

AI USE STATEMENT

In this work, we used generative AI tools for aiding or polishing writing, implementing methods, summarizing or analyzing existing literature, and editing software code. We have not used generative AI tools for proposing or refining hypotheses, assisting in the writing of proofs, assisting with translation, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, designing or providing feedback on research methodology or experiments, supporting qualitative and thematic data analysis, or interpreting results. Generating synthetic data sets, cleaning and reformatting datasets, and transcribing recordings of research material are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. URL https://arxiv.org/abs/2309.16609.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Konstantinos Parasyris, Jiancheng Liu, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. Deepzero: Scaling up zeroth-order optimization for deep model training. In *International Conference on Learning Representations*, volume 2024, pp. 50185–50206, 2024.

Minping Chen, You-Liang Huang, and Zeyi Wen. Towards efficient low-order hybrid optimizer for language model fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23605–23613, 2025a.

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. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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

Krzysztof M Choromanski, Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, and Vikas Sindhwani. From complexity to simplicity: Adaptive es-active subspaces for blackbox optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Sizhe Dang, yangyangGuo, Yanjun Zhao, Xiaodong Zheng, Guang Dai, Ivor Tsang, and Haishan Ye. Fzoo: Fast zeroth-order optimizer for fine-tuning large language models towards adam-scale speed. In *International Conference on Learning Representations*, volume 2026, pp. 113760–113792, 2026.

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

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

Alessio Galatolo, Zhenbang Dai, Katie Winkle, and Meriem Beloucif. Visualising policy-reward interplay to inform zeroth-order preference optimisation of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17446–17461, 2025.

Tanmay Gautam, Youngsuk Park, Hao Zhou, Parameswaran Raman, and Wooseok Ha. Variance-reduced zeroth-order methods for fine-tuning language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 15180–15208. PMLR, 2024.

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

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.

Wentao Guo, Jikai Long, Yimeng Zeng, Zirui Liu, Xinyu Yang, Yide Ran, Jacob Gardner, Osbert Bastani, Christopher De Sa, Xiaodong Yu, Beidi Chen, and Zhaozhuo Xu. Zeroth-order fine-tuning of llms with transferable static sparsity. In *International Conference on Learning Representations*, volume 2025, pp. 14814–14854, 2025.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Shuoran Jiang, Qingcai Chen, Youcheng Pan, Yang Xiang, Yukang Lin, Xiangping Wu, Chuanyi Liu, and Xiaobao Song. Zo-adamu optimizer: Adapting perturbation by the momentum and uncertainty in zeroth-order optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18363–18371, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. 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)*, pp. 4582–4597, 2021.

Zeman Li, Xinwei Zhang, Peilin Zhong, Yuan Deng, Meisam Razaviyayn, and Vahab Mirrokni. Addax: Utilizing zeroth-order gradients to improve memory efficiency and performance of sgd for fine-tuning language models. In *International Conference on Learning Representations*, volume 2025, pp. 880–917, 2025.

Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. volume 31, 2018.

Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Sparse mezo: Less parameters for better performance in zeroth-order llm fine-tuning. *Advances in Neural Information Processing Systems*, 38:21139–21169, 2025.

Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 4264–4273. PMLR, 2019.

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

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

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*, pp. 8577–8595, 2025.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

Hyunseok Seung, Jaewoo Lee, and Hyunsuk Ko. Low-rank curvature for zeroth-order optimization in llm fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25235–25242, 2026.

James C Spall. A stochastic approximation technique for generating maximum likelihood parameter estimates. In *1987 American control conference*, pp. 1161–1167. IEEE, 1987.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Yan Sun, Qixin Zhang, Liang Ding, Li Shen, and Dacheng Tao. Tezo: Empowering the low-rankness on the temporal dimension in the zeroth-order optimization for fine-tuning llms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Shuo Wang, Ziyu Chen, and Ming Tang. Curvzo: Adaptive curvature-guided sparse zeroth-order optimization for efficient llm fine-tuning. In *International Conference on Machine Learning*, 2026.

Ziming Yu, Pan Zhou, Sike Wang, Jia Li, Mi Tian, and Hua Huang. Zeroth-order fine-tuning of llms in random subspaces. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4475–4485. IEEE, 2025.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Kairun Zhang, Haoyu Li, Yanjun Zhao, Yifan Sun, and Huan Zhang. Learning a zeroth-order optimizer for fine-tuning llms. In *International Conference on Machine Learning*, 2026.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

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

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

Jiajun Zhou, Yifan Yang, Kai Zhen, Ziyue Liu, Yequan Zhao, Ershad Banijamali, Athanasios Mouchtaris, Ngai Wong, and Zheng Zhang. Quzo: Quantized zeroth-order fine-tuning for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5341–5359, 2025.

A Algorithmic Details

Algorithms 2 and 3 instantiate the generic procedure of Algorithm 1 for ZO-Adam and LOZO-M. The lines contributed by MAPS are marked in blue. Algorithm 2 samples the dense direction $\boldsymbol{z}$ and replaces it with the guided direction $\boldsymbol{S}$, after which the ZO-Adam moments and the parameter update are computed from $\widehat{\boldsymbol{g}} = c\boldsymbol{S}$. Algorithm 3 samples the left factor $\boldsymbol{U}$ of the low-rank perturbation and replaces it with the guided factor $\boldsymbol{S}$ before the two function evaluations, while the right factor $\boldsymbol{V}$ follows the lazy sampling and subspace projection of LOZO-M.

Algorithm 2 MAPS-ZO-ADAM

Require: Loss $F(\boldsymbol{\theta};\xi)$, trainable parameter blocks $\{\boldsymbol{\theta}_\ell\}_{\ell=1}^{B}$, steps T, perturbation radius μ, learning rate η, momentum coefficients β_1, β_2, constant ϵ and MAPS parameters.

Initialize the Adam first- and second-moment states $\boldsymbol{m}_\ell$ and $\boldsymbol{v}_\ell$ for all ℓ.
for $t = 1, \dots, T$ **do**
  Obtain λ_t and $\mathcal{S}_t$ according to Section 4.2
  for $\ell = 1, \dots, B$ **do**
    Sample $\boldsymbol{z}_{t,\ell} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
    if $\ell \in \mathcal{S}_t$, $\lambda_t > 0$ and $\|\boldsymbol{m}_{t-1,\ell}\|_F > \epsilon$ **then**
      $\boldsymbol{S}_{t,\ell} \leftarrow \sqrt{1-\lambda_t}\boldsymbol{z}_{t,\ell} + \sqrt{\lambda_t \dim(\boldsymbol{\theta}_\ell)} \dfrac{\boldsymbol{m}_{t-1,\ell}}{\max\{\|\boldsymbol{m}_{t-1,\ell}\|_F, \epsilon\}}$
    else $\boldsymbol{S}_{t,\ell} \leftarrow \boldsymbol{z}_{t,\ell}$
  $F_+ \leftarrow F(\boldsymbol{\theta}_t, +\mu, \boldsymbol{S}_t, \xi_t)$; $F_- \leftarrow F(\boldsymbol{\theta}_t, -\mu, \boldsymbol{S}_t, \xi_t)$; $c_t \leftarrow (F_+ - F_-)/(2\mu)$
  for all $\ell = 1, \dots, B$ **do** $\widehat{\boldsymbol{g}}_{t,\ell} \leftarrow c_t \boldsymbol{S}_{t,\ell}$
    $\boldsymbol{m}_{t,\ell} \leftarrow \beta_1 \boldsymbol{m}_{t-1,\ell} + (1-\beta_1)\widehat{\boldsymbol{g}}_{t,\ell}$; $\boldsymbol{v}_{t,\ell} \leftarrow \beta_2 \boldsymbol{v}_{t-1,\ell} + (1-\beta_2)\widehat{\boldsymbol{g}}_{t,\ell}^2$
    $\widehat{\boldsymbol{m}}_{t,\ell} \leftarrow \boldsymbol{m}_{t,\ell}/(1-\beta_1^t)$; $\widehat{\boldsymbol{v}}_{t,\ell} \leftarrow \boldsymbol{v}_{t,\ell}/(1-\beta_2^t)$; $\boldsymbol{\theta}_{t+1,\ell} \leftarrow \boldsymbol{\theta}_{t,\ell} - \eta \dfrac{\widehat{\boldsymbol{m}}_{t,\ell}}{\sqrt{\widehat{\boldsymbol{v}}_{t,\ell}} + \epsilon}$

Algorithm 3 MAPS-LOZO-M

Require: Loss $F(\boldsymbol{X};\xi)$, matrix blocks $\{\boldsymbol{X}_\ell\}_{\ell=1}^{B}$, steps T, perturbation radius μ, learning rate η, rank r, lazy sampling interval ν, momentum coefficient β, constant ϵ and MAPS parameters.

Initialize the momentum states following LOZO-M.
for $t = 1, \dots, T$ **do**
  Obtain λ_t and $\mathcal{S}_t$ according to Section 4.2
  for $\ell = 1, \dots, B$ **do**
    if $(t-1) \bmod \nu = 0$ **then** Sample $\boldsymbol{V}_{t,\ell} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
      if $t > 1$ **then** $\boldsymbol{N}_{t-1,\ell} \leftarrow \boldsymbol{N}_{t-1,\ell}\boldsymbol{V}_{t-1,\ell}^\top \boldsymbol{V}_{t,\ell}/n_\ell$
    else $\boldsymbol{V}_{t,\ell} \leftarrow \boldsymbol{V}_{t-1,\ell}$
    Sample $\boldsymbol{U}_{t,\ell} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
    if $\ell \in \mathcal{S}_t$, $\lambda_t > 0$ and $\|\boldsymbol{N}_{t-1,\ell}\|_F > \epsilon$ **then**
      $\boldsymbol{S}_{t,\ell} \leftarrow \sqrt{1-\lambda_t}\boldsymbol{U}_{t,\ell} + \sqrt{\lambda_t m_\ell r} \dfrac{\boldsymbol{N}_{t-1,\ell}}{\max\{\|\boldsymbol{N}_{t-1,\ell}\|_F, \epsilon\}}$
    else $\boldsymbol{S}_{t,\ell} \leftarrow \boldsymbol{U}_{t,\ell}$
  $F_+ \leftarrow F(\boldsymbol{X}_t, +\mu, \boldsymbol{S}_t, \boldsymbol{V}_t, \xi_t)$; $F_- \leftarrow F(\boldsymbol{X}_t, -\mu, \boldsymbol{S}_t, \boldsymbol{V}_t, \xi_t)$; $c_t \leftarrow (F_+ - F_-)/(2\mu)$
  for all $\ell = 1, \dots, B$ **do** $\boldsymbol{N}_{t,\ell} \leftarrow \beta \boldsymbol{N}_{t-1,\ell} + (1-\beta) c_t \boldsymbol{S}_{t,\ell}$; $\boldsymbol{X}_{t+1,\ell} \leftarrow \boldsymbol{X}_{t,\ell} - \eta \boldsymbol{N}_{t,\ell}\boldsymbol{V}_{t,\ell}^\top / r$

B Convergence Analysis

Without loss of generality, we consider the convergence for one matrix block $\ell = 1$. Let $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, $\boldsymbol{U}_t \in \mathbb{R}^{m\times r}$, and set $d := mr$. $\mathbb{E}_t[\cdot]$ denotes the conditional expectation given all information available before sampling the current activation variables, $\boldsymbol{U}_t$, and ξ_t. So the MAPS direction

following Algorithm 3 is:

$$\boldsymbol{S}_t = \begin{cases} \sqrt{1-\lambda_t}\,\boldsymbol{U}_t + \sqrt{\lambda_t m r}\,\dfrac{\boldsymbol{N}_{t-1}}{\|\boldsymbol{N}_{t-1}\|_F}, & \text{if guidance is activated,} \\ \boldsymbol{U}_t, & \text{otherwise.} \end{cases} \tag{7}$$

where $\boldsymbol{N}_{t-1}$ denotes the momentum after the LOZO-M subspace transport, and $\boldsymbol{U}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ after vectorization. Fix a lazy period k and write $t = k\nu + s, s \in \{0, 1, \ldots, \nu - 1\}$. As in LOZO (Chen et al., 2025b), introduce

$$\boldsymbol{X}_{k\nu+s} = \boldsymbol{X}_{k\nu} + \boldsymbol{B}_k^s (\boldsymbol{V}^{(k)})^\top, \qquad \boldsymbol{B}_k^0 = \boldsymbol{0}. \tag{8}$$

and define the reduced objective

$$G_k(\boldsymbol{B}; \xi) := F\Big(\boldsymbol{X}_{k\nu} + \boldsymbol{B}(\boldsymbol{V}^{(k)})^\top; \xi\Big), \qquad g_k(\boldsymbol{B}) := f\Big(\boldsymbol{X}_{k\nu} + \boldsymbol{B}(\boldsymbol{V}^{(k)})^\top\Big). \tag{9}$$

Under Assumptions 4.1 and 4.2, G_k is $\widetilde{L}$-smooth and its stochastic-gradient variance is bounded by $\widetilde{\sigma}^2$, where

$$\widetilde{L} = nL, \qquad \widetilde{\sigma}^2 = n\sigma^2. \tag{10}$$

At time $t = k\nu + s$, the MAPS-LOZO-M updates can be equivalently expressed in terms of $\boldsymbol{B}_k^s$ as

$$c_t = \frac{G_k(\boldsymbol{B}_k^s + \mu \boldsymbol{S}_t; \xi_t) - G_k(\boldsymbol{B}_k^s - \mu \boldsymbol{S}_t; \xi_t)}{2\mu}, \qquad \widehat{\boldsymbol{g}}_t = c_t \boldsymbol{S}_t, \tag{11}$$

$$\boldsymbol{N}_t = \beta \boldsymbol{N}_{t-1} + (1-\beta)\widehat{\boldsymbol{g}}_t, \qquad \boldsymbol{B}_k^{s+1} = \boldsymbol{B}_k^s - \frac{\eta}{r}\boldsymbol{N}_t. \tag{12}$$

Lemma B.1. *For any $\boldsymbol{A} \in \mathbb{R}^{m\times r}$ independent of the current random sampling,*

$$(1-\widetilde{\lambda})\|\boldsymbol{A}\|_F^2 \le \mathbb{E}_t\langle \boldsymbol{A}, \boldsymbol{S}_t\rangle_F^2 \le [1 + (d-1)\widetilde{\lambda}]\|\boldsymbol{A}\|_F^2, \tag{13}$$

$$\mathbb{E}_t\|\boldsymbol{S}_t\|_F^2 = d, \tag{14}$$

$$\mathbb{E}_t[\langle \boldsymbol{A}, \boldsymbol{S}_t\rangle_F^2 \|\boldsymbol{S}_t\|_F^2] \le (d+4)\mathbb{E}_t\langle \boldsymbol{A}, \boldsymbol{S}_t\rangle_F^2, \tag{15}$$

$$\mathbb{E}_t\|\boldsymbol{S}_t\|_F^6 \le d(d+2)(d+4). \tag{16}$$

Proof. Let $\pi_t \le pq$ be the conditional probability that guidance is applied to the analyzed block. The inequality allows for warmup and an unavailable momentum state. When guidance is activated, according to Equation (7), we have

$$\mathbb{E}_t\langle \boldsymbol{A}, \boldsymbol{S}_t\rangle_F^2 = (1-\lambda)\|\boldsymbol{A}\|_F^2 + \lambda d \left\langle \boldsymbol{A}, \frac{\boldsymbol{N}_{t-1}}{\|\boldsymbol{N}_{t-1}\|_F} \right\rangle_F^2. \tag{17}$$

When guidance is not activated, $\boldsymbol{S}_t = \boldsymbol{U}_t$. Averaging over the two cases gives

$$\mathbb{E}_t\langle \boldsymbol{A}, \boldsymbol{S}_t\rangle_F^2 = (1-\pi_t\lambda)\|\boldsymbol{A}\|_F^2 + \pi_t\lambda d \left\langle \boldsymbol{A}, \frac{\boldsymbol{N}_{t-1}}{\|\boldsymbol{N}_{t-1}\|_F} \right\rangle_F^2. \tag{18}$$

Using the Cauchy–Schwarz inequality and $0 \le \pi_t\lambda \le pq\lambda = \widetilde{\lambda}$, we obtain Equation (13).

Similarly, when guidance is activated, $\mathbb{E}_t\|\boldsymbol{S}_t\|_F^2 = (1-\lambda)\mathbb{E}_t\|\boldsymbol{U}_t\|_F^2 + \lambda d = d$, and the same equality holds when guidance is inactive. Hence, $\mathbb{E}_t\|\boldsymbol{S}_t\|_F^2 = d$.

By applying standard Gaussian moment formulas, we obtain the following bound:

$$\mathbb{E}_t\left[\langle \boldsymbol{A}, \boldsymbol{S}_t\rangle_F^2 \|\boldsymbol{S}_t\|_F^2\right] = (1-\lambda)(d+2-2\lambda)\|\boldsymbol{A}\|_F^2 + \lambda d(d+4-4\lambda)\left\langle \boldsymbol{A}, \frac{\boldsymbol{N}_{t-1}}{\|\boldsymbol{N}_{t-1}\|_F} \right\rangle_F^2. \tag{19}$$

Since $d + 2 - 2\lambda \le d + 4$ and $\lambda d(d+4-4\lambda) \le (d+4)\lambda d$, Equation (15) is proved. For the unguided perturbation, the standard Gaussian gives $\mathbb{E}_t\left[\langle \boldsymbol{A}, \boldsymbol{U}_t\rangle_F^2 \|\boldsymbol{U}_t\|_F^2\right] = (d+2)\|\boldsymbol{A}\|_F^2$, which also satisfies the same upper bound.

Finally, we have $\mathbb{E}_t\|\boldsymbol{S}_t\|_F^6 = d^3 + 6d^2(1-\lambda^2) + 8d(1-\lambda)^2(1+2\lambda) \le d(d+2)(d+4)$ for activated guidance and $\mathbb{E}_t\|\boldsymbol{U}_t\|_F^6 = d(d+2)(d+4)$ for the unguided case, which proves Equation (16).

□

For convenience, define $S_t := \mathbb{E}_t \langle \nabla g_k(\boldsymbol{B}_k^s), \boldsymbol{S}_t \rangle_F^2$.

Lemma B.2. *The MAPS estimator satisfies*

$$\mathbb{E}_t \langle \nabla g_k(\boldsymbol{B}_k^s), \widehat{\boldsymbol{g}}_t \rangle_F \geq \frac{3}{4} S_t - \frac{\widetilde{L}^2 \mu^2 d}{1-\widetilde{\lambda}} \left(1 + \frac{d}{2}\sqrt{\widetilde{\lambda}}\right)^2, \tag{20}$$

$$\mathbb{E}_t \|\widehat{\boldsymbol{g}}_t\|_F^2 \leq 2(d+4) S_t + 2(d+4)[1 + (d-1)\widetilde{\lambda}]\widetilde{\sigma}^2 + \frac{\widetilde{L}^2 \mu^2}{2} d(d+2)(d+4). \tag{21}$$

Proof. For fixed $\boldsymbol{S}_t$, taking expectation over ξ_t replaces the stochastic objective G_k in the symmetric difference by g_k. Applying the standard Gaussian smoothing argument to the random component of $\boldsymbol{S}_t$, together with the $\widetilde{L}$-smoothness of g_k for the momentum-guided component, gives

$$\left\| \mathbb{E}_t[\widehat{\boldsymbol{g}}_t] - \mathbb{E}_t[\langle \nabla g_k(\boldsymbol{B}_k^s), \boldsymbol{S}_t \rangle_F \boldsymbol{S}_t] \right\|_F \leq \widetilde{L}\mu\sqrt{d} \left(1 + \frac{d}{2}\sqrt{\widetilde{\lambda}}\right). \tag{22}$$

Therefore,

$$\mathbb{E}_t \left\langle \nabla g_k(\boldsymbol{B}_k^s), \widehat{\boldsymbol{g}}_t \right\rangle_F \geq S_t - \widetilde{L}\mu\sqrt{d} \left(1 + \frac{d}{2}\sqrt{\widetilde{\lambda}}\right) \left\| \nabla g_k(\boldsymbol{B}_k^s) \right\|_F. \tag{23}$$

By the previous moment bound, $S_t \geq (1-\widetilde{\lambda}) \left\| \nabla g_k(\boldsymbol{B}_k^s) \right\|_F^2$. Substituting it into Equation (23) and using $xy \leq x^2/4 + y^2$ gives Equation (20).

For the second moment, the $\widetilde{L}$-smoothness of G_k implies

$$|c_t - \langle \nabla G_k(\boldsymbol{B}_k^s; \xi_t), \boldsymbol{S}_t \rangle_F| \leq \frac{\widetilde{L}\mu}{2} \left\| \boldsymbol{S}_t \right\|_F^2. \tag{24}$$

Since $\widehat{\boldsymbol{g}}_t = c_t \boldsymbol{S}_t$, we have

$$\begin{aligned} \mathbb{E}_t \left\| \widehat{\boldsymbol{g}}_t \right\|_F^2 = \mathbb{E}_t \left[c_t^2 \left\| \boldsymbol{S}_t \right\|_F^2 \right] &\leq 2\mathbb{E}_t \left[\langle \nabla G_k(\boldsymbol{B}_k^s; \xi_t), \boldsymbol{S}_t \rangle_F^2 \left\| \boldsymbol{S}_t \right\|_F^2 \right] \\ &\quad + 2\mathbb{E}_t \left[\left(c_t - \langle \nabla G_k(\boldsymbol{B}_k^s; \xi_t), \boldsymbol{S}_t \rangle_F \right)^2 \left\| \boldsymbol{S}_t \right\|_F^2 \right] \\ &\leq 2\mathbb{E}_t \left[\langle \nabla G_k(\boldsymbol{B}_k^s; \xi_t), \boldsymbol{S}_t \rangle_F^2 \left\| \boldsymbol{S}_t \right\|_F^2 \right] + \frac{\widetilde{L}^2 \mu^2}{2} \mathbb{E}_t \left\| \boldsymbol{S}_t \right\|_F^6. \end{aligned} \tag{25}$$

According to $\nabla G_k(\boldsymbol{B}_k^s; \xi_t) = \nabla g_k(\boldsymbol{B}_k^s) + [\nabla G_k(\boldsymbol{B}_k^s; \xi_t) - \nabla g_k(\boldsymbol{B}_k^s)]$, we have

$$\begin{aligned} \mathbb{E}_t \left[\langle \nabla G_k(\boldsymbol{B}_k^s; \xi_t), \boldsymbol{S}_t \rangle_F^2 \left\| \boldsymbol{S}_t \right\|_F^2 \right] &= \mathbb{E}_t \left[\langle \nabla g_k(\boldsymbol{B}_k^s), \boldsymbol{S}_t \rangle_F^2 \left\| \boldsymbol{S}_t \right\|_F^2 \right] \\ &\quad + \mathbb{E}_t \left[\langle \nabla G_k(\boldsymbol{B}_k^s; \xi_t) - \nabla g_k(\boldsymbol{B}_k^s), \boldsymbol{S}_t \rangle_F^2 \left\| \boldsymbol{S}_t \right\|_F^2 \right] \\ &\leq (d+4) S_t + (d+4) \left[1 + (d-1)\widetilde{\lambda} \right] \widetilde{\sigma}^2. \end{aligned} \tag{26}$$

Combining Equation (26), Equation (16) and Equation (25) gives Equation (21).

□

Lemma B.3. *The subspace projection does not increase the expected squared norm of the momentum.If the momentum is initialized to zero, the following bounds hold.*

$$\sum_{t=0}^{T-1} \mathbb{E} \| \boldsymbol{N}_t (\boldsymbol{V}^{(k(t))})^\top \|_F^2 \leq n \sum_{t=0}^{T-1} \mathbb{E} \| \widehat{\boldsymbol{g}}_t \|_F^2. \tag{27}$$

Proof. At a subspace refresh, Assumption 4.2 and independence of the new right factor imply

$$\mathbb{E} \left\| \frac{1}{n} \boldsymbol{N}_{t-1} (\boldsymbol{V}^{(k-1)})^\top \boldsymbol{V}^{(k)} \right\|_F^2 = \frac{r}{n} \mathbb{E} \| \boldsymbol{N}_{t-1} \|_F^2 \leq \mathbb{E} \| \boldsymbol{N}_{t-1} \|_F^2. \tag{28}$$

Combining projection contraction with convexity of the squared norm gives

$$\mathbb{E}\|\boldsymbol{N}_t\|_F^2 \le \beta\mathbb{E}\|\boldsymbol{N}_{t-1}\|_F^2 + (1-\beta)\mathbb{E}\|\widehat{\boldsymbol{g}}_t\|_F^2. \tag{29}$$

Unrolling the above inequality and using $\boldsymbol{N}_{-1} = \boldsymbol{0}$ yield

$$\mathbb{E}\|\boldsymbol{N}_t\|^2 \le (1-\beta)\sum_{i=0}^{t}\beta^{t-i}\mathbb{E}\|\widehat{\boldsymbol{g}}_i\|^2. \tag{30}$$

Summing over $t = 0, \dots, T-1$ gives

$$\sum_{t=0}^{T-1}\mathbb{E}\|\boldsymbol{N}_t\|^2 \le \sum_{t=0}^{T-1}(1-\beta)\sum_{i=0}^{t}\beta^{t-i}\mathbb{E}\|\widehat{\boldsymbol{g}}_i\|^2 = \sum_{i=0}^{T-1}\left((1-\beta)\sum_{t=i}^{T-1}\beta^{t-i}\right)\mathbb{E}\|\widehat{\boldsymbol{g}}_i\|^2 \le \sum_{i=0}^{T-1}\mathbb{E}\|\widehat{\boldsymbol{g}}_i\|^2. \tag{31}$$

Since $\|\boldsymbol{N}\boldsymbol{V}^\top\|_F^2 = n\|\boldsymbol{N}\|_F^2$, the lemma follows.

□

Lemma B.4. *The expected objective change between two consecutive iterations can be bounded as follows.*

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{X}_{t+1}) - f(\boldsymbol{X}_t)] \le{}& \beta\rho_t\mathbb{E}[f(\boldsymbol{X}_t) - f(\boldsymbol{X}_{t-1})] - \frac{(1-\beta)\eta}{2r}\mathbb{E}[S_t] \\ &+ \frac{L\beta\eta^2}{r^2}\mathbb{E}\|\boldsymbol{N}_{t-1}(\boldsymbol{V}^{(k(t-1))})^\top\|_F^2 + \frac{(1-\beta)\eta\widetilde{L}^2\mu^2 d}{r(1-\widetilde{\lambda})}\left(1 + \frac{d}{2}\sqrt{\widetilde{\lambda}}\right)^2 \\ &+ \frac{Ln(1-\beta)\eta^2(d+4)}{r^2}[1 + (d-1)\widetilde{\lambda}]\widetilde{\sigma}^2 + \frac{Ln(1-\beta)\eta^2\widetilde{L}^2\mu^2}{4r^2}d(d+2)(d+4).\end{aligned} \tag{32}$$

where $\eta \le r/[4Ln(d+4)]$, $\rho_t = r/n$ at a subspace refresh and $\rho_t = 1$ otherwise. And for the initial step, $\boldsymbol{X}_{-1} = \boldsymbol{X}_0$ and $\boldsymbol{N}_{-1} = \boldsymbol{0}$.

Proof. Apply L-smoothness to $\boldsymbol{X}_{t+1} - \boldsymbol{X}_t = -(\eta/r)\boldsymbol{N}_t(\boldsymbol{V}^{(k(t))})^\top$, obtaining

$$\mathbb{E}[f(\boldsymbol{X}_{t+1}) - f(\boldsymbol{X}_t)] \le -\frac{\eta}{r}\mathbb{E}\left\langle\nabla f(\boldsymbol{X}_t), \boldsymbol{N}_t(\boldsymbol{V}^{(k(t))})^\top\right\rangle_F + \frac{L\eta^2}{2r^2}\mathbb{E}\left\|\boldsymbol{N}_t(\boldsymbol{V}^{(k(t))})^\top\right\|_F^2. \tag{33}$$

For the first term, decompose $\boldsymbol{N}_t = \beta\boldsymbol{N}_{t-1} + (1-\beta)\widehat{\boldsymbol{g}}_t$. At a subspace refresh, the independence of the new right factor and Assumption 4.2 imply that the expected full-space representation of the previous momentum is scaled by $\rho_t = r/n$. Otherwise, $\rho_t = 1$. Hence, using $\boldsymbol{X}_t - \boldsymbol{X}_{t-1} = -\frac{\eta}{r}\boldsymbol{N}_{t-1}(\boldsymbol{V}^{(k(t-1))})^\top$, the previous momentum is bounded by

$$\begin{aligned}-\frac{\beta\eta}{r}\mathbb{E}\left\langle\nabla f(\boldsymbol{X}_t), \boldsymbol{N}_{t-1}(\boldsymbol{V}^{(k(t))})^\top\right\rangle_F &= \beta\rho_t\,\mathbb{E}\left\langle\nabla f(\boldsymbol{X}_t), \boldsymbol{X}_t - \boldsymbol{X}_{t-1}\right\rangle_F \\ &\le \beta\rho_t\mathbb{E}[f(\boldsymbol{X}_t) - f(\boldsymbol{X}_{t-1})] + \frac{L\beta\eta^2}{2r^2}\mathbb{E}\|\boldsymbol{N}_{t-1}(\boldsymbol{V}^{(k(t-1))})^\top\|_F^2.\end{aligned} \tag{34}$$

By Equation (20), the current estimator satisfies

$$-\frac{(1-\beta)\eta}{r}\mathbb{E}\left\langle\nabla f(\boldsymbol{X}_t), \widehat{\boldsymbol{g}}_t(\boldsymbol{V}^{(k(t))})^\top\right\rangle_F \le -\frac{3(1-\beta)\eta}{4r}\mathbb{E}[S_t] + \frac{(1-\beta)\eta\widetilde{L}^2\mu^2 d}{r(1-\widetilde{\lambda})}\left(1 + \frac{d}{2}\sqrt{\widetilde{\lambda}}\right)^2. \tag{35}$$

For the second term, apply the convexity of the squared Frobenius norm yielding

$$\mathbb{E}\left\|\boldsymbol{N}_t(\boldsymbol{V}^{(k(t))})^\top\right\|_F^2 \le \beta\mathbb{E}\left\|\boldsymbol{N}_{t-1}(\boldsymbol{V}^{(k(t-1))})^\top\right\|_F^2 + n(1-\beta)\mathbb{E}\left\|\widehat{\boldsymbol{g}}_t\right\|_F^2. \tag{36}$$

We then apply Equation (21) to obtain its upper bound. Thus, combining the two terms and using $\eta \le r/[4Ln(d+4)]$ gives

$$-\frac{3(1-\beta)\eta}{4r}\mathbb{E}S_t + \frac{Ln(1-\beta)\eta^2(d+4)}{r^2}\mathbb{E}S_t \le -\frac{(1-\beta)\eta}{2r}\mathbb{E}S_t. \tag{37}$$

Then Equation (32) follows by combining the above bounds. □

Lemma B.5. *Let $T = K\nu$ and initialize the momentum to zero. Then*

$$\sum_{k=0}^{K-1}\sum_{s=0}^{\nu-1}\mathbb{E}\|\boldsymbol{B}_k^s\|_F^2 \le \frac{3\eta^2\nu}{r^2(1-\beta)^2}\sum_{t=0}^{T-1}\mathbb{E}\|\widehat{\boldsymbol{g}}_t\|_F^2 + \frac{6\eta^2\nu^2}{r^2}[1+(d-1)\widetilde{\lambda}]\sum_{t=0}^{T-1}\mathbb{E}[S_t] + \frac{6\eta^2\nu^2 T\widetilde{L}^2\mu^2 d}{r^2}\left(1+\frac{d}{2}\sqrt{\widetilde{\lambda}}\right)^2.$$

Proof. Within period k, unrolling the momentum updates and $\boldsymbol{B}_k^{s+1} = \boldsymbol{B}_k^s - \frac{\eta}{r}\boldsymbol{N}_{k\nu+s}$ gives

$$\boldsymbol{B}_k^s = -\frac{\eta}{r}\left[\frac{\beta(1-\beta^s)}{1-\beta}\boldsymbol{N}_{k\nu-1} + \sum_{j=0}^{s-1}(1-\beta^{s-j})\widehat{\boldsymbol{g}}_{k\nu+j}\right], \tag{38}$$

where $\boldsymbol{N}_{k\nu-1}$ denotes the boundary momentum projected onto the current subspace, and $\boldsymbol{N}_{-1} = \boldsymbol{0}$.

Write $\boldsymbol{h}_t = \mathbb{E}_t\widehat{\boldsymbol{g}}_t$. Since S_t is induced by the same second-moment operator as in Equation (13). The largest eigenvalue is at most $1+(d-1)\widetilde{\lambda}$, we have

$$\|\mathbb{E}_t[\langle\nabla g_k(\boldsymbol{B}_k^s), \boldsymbol{S}_t\rangle_F \boldsymbol{S}_t]\|_F^2 \le [1+(d-1)\widetilde{\lambda}]S_t. \tag{39}$$

Combining this with the estimator bias bound yields

$$\|\boldsymbol{h}_t\|_F^2 \le 2[1+(d-1)\widetilde{\lambda}]S_t + 2\widetilde{L}^2\mu^2 d\left(1+\frac{d}{2}\sqrt{\widetilde{\lambda}}\right)^2. \tag{40}$$

Decompose the estimator as $\widehat{\boldsymbol{g}}_t = \boldsymbol{h}_t + (\widehat{\boldsymbol{g}}_t - \boldsymbol{h}_t)$. For $i<j$, $\mathbb{E}\langle\widehat{\boldsymbol{g}}_i - \boldsymbol{h}_i,\ \widehat{\boldsymbol{g}}_j - \boldsymbol{h}_j\rangle_F = 0$. Applying $\|a+b+c\|^2 \le 3(\|a\|^2+\|b\|^2+\|c\|^2)$, $1-\beta^{s-j}\le 1$ and Cauchy–Schwarz to the unrolled updates, we obtain

$$\mathbb{E}\|\boldsymbol{B}_k^s\|_F^2 \le \frac{3\eta^2}{r^2}\left[\frac{\beta^2}{(1-\beta)^2}\mathbb{E}\|\boldsymbol{N}_{k\nu-1}\|_F^2 + s\sum_{j=0}^{s-1}\mathbb{E}\|\boldsymbol{h}_{k\nu+j}\|_F^2 + \sum_{j=0}^{s-1}\mathbb{E}\|\widehat{\boldsymbol{g}}_{k\nu+j}\|_F^2\right]. \tag{41}$$

Then summing over $k=0,\dots,K-1$ and $s=0,\dots,\nu-1$ yields

$$\sum_{k,s}\mathbb{E}\|\boldsymbol{B}_k^s\|_F^2 \le \frac{3\eta^2}{r^2}\left[\frac{\nu\beta^2}{(1-\beta)^2}\sum_k\mathbb{E}\|\boldsymbol{N}_{k\nu-1}\|_F^2 + \nu^2\sum_t\mathbb{E}\|\boldsymbol{h}_t\|_F^2 + \nu\sum_t\mathbb{E}\|\widehat{\boldsymbol{g}}_t\|_F^2\right]. \tag{42}$$

At the beginning of a new lazy period, the subspace projection does not increase the expected squared norm of the previous momentum. Together with Lemma B.3 and $1+\beta^2/(1-\beta)^2 \le (1-\beta)^{-2}$, this lemma is proved.

□

Theorem B.6 (Convergence of MAPS-LOZO-M)**.** *Suppose Assumptions 4.1 and 4.2 hold, $\beta\in[0,1)$, and $\widetilde{\lambda}<1$. Let $T = K\nu$ and $\Delta_0 > 0$. If we choose*

$$\eta = \left(\frac{1024Lmn\nu(1+d\widetilde{\lambda})}{(1-\beta)^2} + \sqrt{\frac{TLmn^2\sigma^2(1+\beta)(1+d\widetilde{\lambda})}{\Delta_0 r(1-\beta)}}\right)^{-1}, \tag{43}$$

and

$$\mu^2 \le \frac{\Delta_0(1-\widetilde{\lambda})\nu}{TLmnr(1+d\sqrt{\widetilde{\lambda}})^2}. \tag{44}$$

Then it holds that:

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla f(\boldsymbol{X}_{k\nu})\|_F^2 \le \frac{80\sqrt{(1+\beta)(1+d\widetilde{\lambda})}}{(1-\beta)^{3/2}(1-\widetilde{\lambda})}\sqrt{\frac{\Delta_0 Lmn^2\sigma^2}{rT}} + \frac{16384(1+d\widetilde{\lambda})}{(1-\beta)^3(1-\widetilde{\lambda})}\frac{\Delta_0 Lmn\nu}{T}. \tag{45}$$

Proof. By Equation (43), $\eta \le \frac{(1-\beta)^2}{1024Lmn\nu(1+d\widetilde{\lambda})} \le \frac{r}{4Ln(d+4)}$, so Lemma B.4 applies. Expanding its recursion and summing over $t = 0, \dots, T-1$, using $0 \le \beta\rho_t \le \beta < 1$, $\sum_{j=0}^{\infty} \beta^j = (1-\beta)^{-1}$, Lemma B.3, and $f \ge f_\star$, gives

$$\begin{aligned}\frac{(1-\beta)\eta}{2r}\sum_{t=0}^{T-1}\mathbb{E}[S_t] \le \Delta_0 + \frac{\widetilde{L}\beta\eta^2}{r^2(1-\beta)}\sum_{t=0}^{T-1}\mathbb{E}\|\widehat{\boldsymbol{g}}_t\|_F^2 + \frac{\eta T\widetilde{L}^2\mu^2 d}{r(1-\widetilde{\lambda})}\left(1+\frac{d}{2}\sqrt{\widetilde{\lambda}}\right)^2 \\ + \frac{T\widetilde{L}\eta^2(d+4)}{r^2}[1+(d-1)\widetilde{\lambda}]\widetilde{\sigma}^2 + \frac{T\widetilde{L}^3\eta^2\mu^2}{4r^2}d(d+2)(d+4).\end{aligned} \tag{46}$$

Substitute Equation (21) from Lemma B.2 into Equation (46). Since $\frac{2\widetilde{L}\beta\eta^2(d+4)}{r^2(1-\beta)} \le \frac{(1-\beta)\eta}{4r}$, moving the resulting multiple of $\sum_t \mathbb{E}[S_t]$ to the left-hand side yields

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[S_t] \le \frac{4r\Delta_0}{(1-\beta)\eta T} + \frac{4\widetilde{L}\eta(1+\beta)(d+4)[1+(d-1)\widetilde{\lambda}]}{r(1-\beta)^2}\widetilde{\sigma}^2 \\ + \frac{4\widetilde{L}^2\mu^2 d}{(1-\beta)(1-\widetilde{\lambda})}\left(1+\frac{d}{2}\sqrt{\widetilde{\lambda}}\right)^2 + \frac{\widetilde{L}^3\eta\mu^2(1+\beta)}{r(1-\beta)^2}d(d+2)(d+4).\end{aligned} \tag{47}$$

For $t = k\nu + s$, Lemma B.1 and the $\widetilde{L}$-smoothness of g_k imply

$$\mathbb{E}\|\nabla g_k(0)\|_F^2 \le \frac{2}{1-\widetilde{\lambda}}\mathbb{E}[S_t] + 2\widetilde{L}^2\mathbb{E}\|\boldsymbol{B}_k^s\|_F^2. \tag{48}$$

Moreover, Assumption 4.2 and independence of $\boldsymbol{V}^{(k)}$ from $\boldsymbol{X}_{k\nu}$ give

$$\mathbb{E}\|\nabla g_k(0)\|_F^2 = \mathbb{E}\|\nabla f(\boldsymbol{X}_{k\nu})\boldsymbol{V}^{(k)}\|_F^2 = r\,\mathbb{E}\|\nabla f(\boldsymbol{X}_{k\nu})\|_F^2. \tag{49}$$

Averaging over k, s and applying Lemmas B.5 and B.2, we obtain

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla f(\boldsymbol{X}_{k\nu})\|_F^2 \le \frac{3}{r(1-\widetilde{\lambda})T}\sum_{t=0}^{T-1}\mathbb{E}[S_t] + \frac{12\widetilde{L}^2\eta^2\nu(d+4)[1+(d-1)\widetilde{\lambda}]}{r^3(1-\beta)^2}\widetilde{\sigma}^2 \\ + \frac{3\widetilde{L}^4\eta^2\nu\mu^2}{r^3(1-\beta)^2}d(d+2)(d+4) + \frac{12\widetilde{L}^4\eta^2\nu^2\mu^2 d}{r^3}\left(1+\frac{d}{2}\sqrt{\widetilde{\lambda}}\right)^2.\end{aligned} \tag{50}$$

Here Equation (43) ensures that

$$\frac{2}{r(1-\widetilde{\lambda})} + \frac{12\widetilde{L}^2\eta^2\nu(d+4)}{r^3(1-\beta)^2} + \frac{12\widetilde{L}^2\eta^2\nu^2[1+(d-1)\widetilde{\lambda}]}{r^3} \le \frac{3}{r(1-\widetilde{\lambda})}. \tag{51}$$

Substituting Equation (47), using $d+4 \le 5d$, $d+2 \le 3d$, $d = mr$, $\widetilde{L} = nL$, and $\widetilde{\sigma}^2 = n\sigma^2$, gives

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla f(\boldsymbol{X}_{k\nu})\|_F^2 \le \frac{12\Delta_0}{(1-\beta)(1-\widetilde{\lambda})T\eta} \\ + \frac{64(1+\beta)(1+d\widetilde{\lambda})}{(1-\beta)^2(1-\widetilde{\lambda})}\frac{Lmn^2\sigma^2\eta}{r} + \frac{4(1+d\sqrt{\widetilde{\lambda}})^2}{(1-\beta)(1-\widetilde{\lambda})^2}\frac{Lmnr\mu^2}{\nu\eta}.\end{aligned} \tag{52}$$

The constants in Equation (52) follow from Equation (43). By Equation (44), the first and third terms combine to give $\frac{16\Delta_0}{(1-\beta)(1-\widetilde{\lambda})T\eta}$. With the choices of μ and η, substitution yields Equation (45).

□

Corollary B.7 (Recovery of LOZO-M)**.** *If $\widetilde{\lambda} = 0$, MAPS-LOZO-M reduces to LOZO-M. With the corresponding choices in Equation (43) and Equation (44),*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla f(\boldsymbol{X}_{k\nu})\|_F^2 \le \frac{80\sqrt{1+\beta}}{(1-\beta)^{3/2}}\sqrt{\frac{\Delta_0 Lmn^2\sigma^2}{rT}} + \frac{16384\Delta_0 Lmn\nu}{(1-\beta)^3 T}. \tag{53}$$

Proof. Setting $\lambda = 0$, $p = 0$, or $q = 0$ gives $\boldsymbol{S}_t = \boldsymbol{U}_t$. The momentum projection and subsequent updates are unchanged, so the algorithm reduces to LOZO-M. Substituting $\widetilde{\lambda} = 0$ into Equation (45) proves the bound. □

Corollary B.8 (Recovery of LOZO)**.** *If $\widetilde{\lambda} = 0$ and $\beta = 0$, MAPS-LOZO-M reduces to LOZO, and*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\|\nabla f(\boldsymbol{X}_{k\nu})\|_F^2 \le 80\sqrt{\frac{\Delta_0 L m n^2 \sigma^2}{rT}} + \frac{16384\Delta_0 L m n \nu}{T}. \tag{54}$$

This recovers the convergence order of LOZO (Chen et al., 2025b).

Proof. With guidance disabled and $\beta = 0$, $\boldsymbol{N}_t = \widehat{\boldsymbol{g}}_t = c_t \boldsymbol{U}_t$, and the parameter update is exactly the LOZO update. The result follows from Corollary B.7 by setting $\beta = 0$. □

C Additional Experiments

C.1 Computational Overhead

We measure peak allocated GPU memory and per-iteration wall-clock time for each MAPS instance and its base optimizer on OPT-13B under the same hardware setting. Table 7 reports the raw measurements on SST-2, MultiRC, and SQuAD. Within every pair, the peak memory is identical, which is consistent with MAPS reusing the existing momentum state and keeping the same two-point query budget. The additional cost is confined to normalizing the momentum and mixing it with the random perturbation. Relative to the corresponding base optimizer, this increases the per-iteration wall-clock time by 4.6%–6.3% for MAPS-LOZO-M, 3.7%–4.3% for MAPS-ZO-ADAM, and 1.4%–2.6% for MAPS-ZO-ADAM with LoRA. The lower overhead under LoRA is expected, as mixing is restricted to the adapter parameters, while forward passes dominate the step runtime.

Table 7: Peak GPU memory and per-iteration wall-clock time on OPT-13B. Memory is identical within each pair. Δ is the relative time increase of MAPS over its base optimizer.

	SST-2			MultiRC			SQuAD		
Method	Mem. (GB)	Time (ms)	Δ	Mem. (GB)	Time (ms)	Δ	Mem. (GB)	Time (ms)	Δ
LOZO-M	24.50	373.5	–	24.50	487.9	–	24.50	384.0	–
MAPS-LOZO-M		390.8	+4.6%		512.2	+5.0%		408.2	+6.3%
ZO-Adam	71.92	1064.1	–	72.28	1183.2	–	71.97	1080.2	–
MAPS-ZO-ADAM		1109.6	+4.3%		1227.4	+3.7%		1126.4	+4.3%
ZO-Adam (LoRA)	24.04	113.6	–	24.44	225.1	–	24.13	123.7	–
MAPS-ZO-ADAM (LoRA)		115.8	+1.9%		228.2	+1.4%		126.9	+2.6%

C.2 Sensitivity to p and $\lambda_{\max}$

We study the sensitivity of MAPS-LOZO-M to the block ratio p and the maximum guidance strength $\lambda_{\max}$ on WiC with OPT-1.3B. It uses the same training configuration as the ablation study. Table 8 reports accuracy and training loss when p is varied at $\lambda_{\max} = 0.3$, and when $\lambda_{\max}$ is varied at $p = 0.5$.

D Implementation Details

D.1 The Hyperparameter Settings

Table 9 summarizes the hyperparameters used for MAPS and its corresponding base optimizers reported in Table 1. Following the experimental setting of Fang et al. (2026), we include Zero-shot, In-Context Learning, full parameter fine-tuning with Adam (Kingma & Ba, 2015), LoRA

Table 8: Sensitivity of MAPS-LOZO-M to the block ratio p and guidance strength $\lambda_{\max}$ on WiC with OPT-1.3B. When varying one hyperparameter, the other is kept fixed.

Block ratio p ($\lambda_{\max} = 0.3$)			Guidance strength $\lambda_{\max}$ ($p = 0.5$)		
p	Acc. (%) ↑	Loss ↓	$\lambda_{\max}$	Acc. (%) ↑	Loss ↓
0.3	59.56	0.66614	0.3	**60.50**	0.66523
0.5	**60.50**	**0.66523**	0.5	59.87	**0.66304**
0.7	57.52	0.66611	0.7	59.09	0.66920
1.0	59.87	0.66735	1.0	60.19	0.66873

fine-tuning (Hu et al., 2022), MeZO and MeZO (LoRA) (Malladi et al., 2023), ZO-AdaMU and ZO-AdaMU (LoRA) (Jiang et al., 2024), LOZO-M (Chen et al., 2025b) and ZO-Adam as baselines. We train all ZO methods for 20000 steps, while the FT methods for 2000 steps. For LoRA, r is the adapter rank and α is the LoRA scaling hyperparameter. All MAPS runs use a fixed warmup ratio $\rho = 0.05$, so the first $\lfloor 0.05T \rfloor$ steps use the random sampler, and a fixed activation probability $q = 0.5$. After warmup, $b_t \sim \mathrm{Bernoulli}(0.5)$ and $\lambda_t \in \{0, \lambda_{\max}\}$ with equal probability. We search only $\lambda_{\max}$ and p; ρ and q are fixed.

Table 9: Hyperparameter settings for MAPS and its base optimizers on OPT-13B.

Experiment	**Hyperparameter**	**Values**
LOZO-M	Batch size	16
	Learning rate	{1e-7, 1e-6}
	Perturbation radius	{1e-4, 1e-3}
	Rank	{2, 4, 8}
	Lazy sampling interval	{50, 100, 200}
	Momentum coefficient	0.9
MAPS-LOZO-M	Batch size	16
	Learning rate	{1e-7, 1e-6}
	Perturbation radius	{1e-4, 1e-3}
	Rank	{2, 4, 8}
	Lazy sampling interval	{50, 100, 200}
	Largest guidance strength $\lambda_{\max}$	{0.1, 0.3, 0.5}
	Block ratio p	{0.5, 0.7, 0.9}
	Momentum coefficient	0.9
ZO-Adam	Batch size	16
	Learning rate	{1e-6, 1e-5, 1e-4, 1e-3}
	Perturbation radius	1e-3
	Momentum coefficients (β_1, β_2)	(0.9, 0.999)
MAPS-ZO-ADAM	Batch size	16
	Learning rate	{1e-6, 1e-5, 1e-4, 1e-3}
	Perturbation radius	1e-3
	Momentum coefficients (β_1, β_2)	(0.9, 0.999)
	Largest guidance strength $\lambda_{\max}$	{0.1, 0.3, 0.5}
	Block ratio p	{0.5, 0.7, 0.9}
ZO-Adam (LoRA)	Batch size	16
	Learning rate	{1e-6, 1e-5, 1e-4, 1e-3}
	Perturbation radius	1e-3
	Momentum coefficients (β_1, β_2)	(0.9, 0.999)
	LoRA configuration (r, α)	(8, 16)
MAPS-ZO-ADAM (LoRA)	Batch size	16
	Learning rate	{1e-6, 1e-5, 1e-4, 1e-3}
	Perturbation radius	1e-3
	Momentum coefficients (β_1, β_2)	(0.9, 0.999)
	LoRA configuration (r, α)	(8, 16)
	Largest guidance strength $\lambda_{\max}$	{0.1, 0.3, 0.5}
	Block ratio p	{0.5, 0.7, 0.9}

D.2 HYPERPARAMETERS FOR OPT-1.3B, QWEN2.5-1.5B AND LLAMA-3.2-1B

Table 10 summarizes the hyperparameter grids used for the experiments in Table 3. All three models share the same search space and are trained for 10000 steps.

Table 10: Hyperparameter grids shared by OPT-1.3B (Zhang et al., 2022), Qwen2.5-1.5B (Qwen et al., 2025), and Llama-3.2-1B (Grattafiori et al., 2024).

Method	Hyperparameter	Values
LOZO-M	Batch size	$\{2, 16\}$
	Learning rate	{1e-7, 1e-6}
	Perturbation radius	{1e-4, 1e-3, 1e-2}
	Rank	$\{2, 4\}$
	Lazy sampling interval	50
MAPS-LOZO-M	Batch size	$\{2, 16\}$
	Learning rate	{1e-7, 1e-6}
	Perturbation radius	{1e-4, 1e-3, 1e-2}
	Rank	$\{2, 4\}$
	Lazy sampling interval	50
	Largest guidance strength $\lambda_{\max}$	$\{0.1, 0.3, 0.5\}$
	Block ratio p	$\{0.5, 0.7\}$

D.3 PROMPT TEMPLATES

Table 11 summarizes the prompt construction used in our OPT experiments, following prior work (Malladi et al., 2023; Jiang et al., 2024).

Table 11: The prompt templates used for the OPT experiments.

Dataset	Type	Prompt
SST-2	CLS	`<sentence> It was`
RTE	CLS	`<premise>` `Does this mean that "<hypothesis>" is true? Yes or No?`
CB	CLS	`Suppose <premise> Can we infer that "<hypothesis>"? Yes, No, or Maybe?`
BoolQ	CLS	`<passage> <Question>?`

WiC	CLS	`Does the word "<word>" have the same meaning in these two sentences? Yes, No?`
		`<sentence1>`
		`<sentence2>`

MultiRC	CLS	`<paragraph>`
		`Question: <question>`
		`I found this answer "<answer>". Is that correct? Yes or No?`

COPA	MC	`<premise> so/because <candidate>`
ReCoRD	MC	`<passage>` `<query>.replace("@placeholder", <candidate>)`

SQuAD	QA	`Title: <title>`
		`Context: <context>`
		`Question: <question>`
		`Answer:`

DROP	QA	`Passage: <passage>`
		`Question: <question>`
		`Answer:`