Universally Empowering Zeroth-Order Optimization via Adaptive Layer-wise Sampling

Fei Wang[1,2], Li Shen[3,5,*], Liang Ding[4], Chao Xue[2], Ye Liu[1], Changxing Ding[1,*]

[1]South China University of Technology, [2]JD Explore Academy,
[3]Shenzhen Campus of Sun Yat-sen University, [4]University of Sydney,
[5]Center for AI Theoretical Foundation and Systems, Shenzhen Loop Area Institute

ft_feiw@mail.scut.edu.cn, chxding@scut.edu.cn

Abstract

Zeroth-Order optimization presents a promising memory-efficient paradigm for fine-tuning Large Language Models by relying solely on forward passes. However, its practical adoption is severely constrained by slow wall-clock convergence and high estimation variance. In this work, we dissect the runtime characteristics of ZO algorithms and identify a critical system bottleneck where the generation of perturbations and parameter updates accounts for over 40% of the training latency. We argue that the standard uniform exploration strategy is fundamentally flawed as it fails to account for the heterogeneous sensitivity of layers in deep networks, resulting in computationally wasteful blind searches. To address this structural mismatch, we propose **AdaLeZO**, an **Ada**ptive **L**ayer-wis**e** **ZO** optimization framework. By formulating the layer selection process as a non-stationary Multi-Armed Bandit problem, AdaLeZO dynamically allocates the limited perturbation budget to the most sensitive parameters. We further introduce an Inverse Probability Weighting mechanism based on sampling with replacement, which guarantees unbiased gradient estimation while effectively acting as a temporal denoiser to reduce variance. Extensive experiments on LLaMA and OPT models ranging from 6.7B to 30B parameters demonstrate that AdaLeZO achieves 1.7× to 3.0× wall-clock acceleration compared to state-of-the-art methods. Crucially, AdaLeZO functions as a universal plug-and-play module that seamlessly enhances the efficiency of existing ZO optimizers without incurring additional memory overhead. [1]

1 Introduction

Large Language Models (LLMs) (Bai et al., 2023; OpenAI et al., 2024; DeepSeek-AI et al., 2024; Grattafiori et al., 2024; Cai et al., 2026) have demonstrated exceptional generalization capabilities across a broad spectrum of natural language processing tasks (Liang et al., 2024; Zhu et al., 2024). To adapt these general-purpose models to specialized downstream domains, Full Fine-Tuning (FFT) remains the gold standard for achieving optimal performance (Rao et al., 2024, 2025). Nevertheless, FFT imposes prohibitive memory requirements as it necessitates the storage of optimizer states and gradient histories (Kingma and Ba, 2015). While Parameter-Efficient Fine-Tuning methods (Zhao et al., 2024; Hu et al., 2022; Li and Liang, 2021), such as LoRA (Hu et al., 2022) and Prefix-Tuning (Li and Liang, 2021), significantly reduce the number of trainable parameters, they still rely on backpropagation. Consequently, these methods require the transient storage of massive intermediate activations proportional to the network depth, which renders the fine-tuning of billion-scale models on consumer-grade hardware computationally infeasible.

To circumvent the memory bottleneck intrinsic to backpropagation, Zeroth-Order (ZO) optimization has recently garnered renewed interest (Chen et al., 2024; Wang et al., 2024; Zhang et al., 2024). By estimating gradients through finite differences using only two forward passes, ZO methods such as MeZO (Malladi et al., 2023) successfully compress the memory footprint of training to inference levels. Despite this breakthrough, standard ZO optimization is plagued by the "curse of dimensionality," where the variance of gradient estimates scales linearly with the parameter size, leading to sluggish convergence and instability (Chen et al., 2025; Sun et al., 2025). Prior research has primarily focused on algorithmic refinements, such as introducing momentum or subspace constraints, to mitigate high variance (Chen et al., 2019; Jiang et al., 2024; Chen et al., 2025; Yu et al., 2025; Zhao et al., 2025). However, these approaches often over-

*Corresponding author.

[1]The official implementation is available at https://github.com/WangFei-2019/UnifiedZO.

Findings of the Association for Computational Linguistics: ACL 2026, pages 40218–40239
July 2-7, 2026

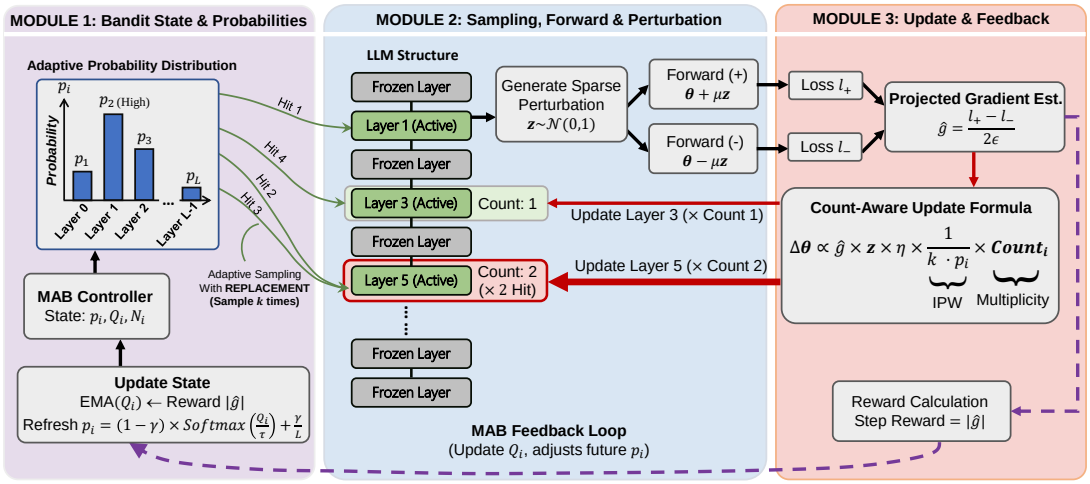


Figure 1: The AdaLeZO workflow. AdaLeZO overcomes the computational inefficiency of uniform ZO exploration by employing the MAB framework to allocate sparse perturbations to sensitive layers adaptively. Key modules include: (1) dynamic layer selection guided by real-time MAB statistics, (2) adaptive sampling with replacement to concentrate the perturbation budget on critical layers, and (3) a count-aware IPW update formula that ensures unbiased gradient estimation and enables efficient, feedback-driven optimization.

look the significant wall-clock latency induced by high-dimensional operations.

In this work, we revisit ZO optimization from two complementary perspectives to uncover the root causes of its inefficiency. From a systems perspective, we observe that the operations required for perturbation generation and parameter updates incur a linear time complexity. Our breakdown analysis on an OPT-6.7B model reveals that these operations constitute nearly half of the per-step training time, creating a substantial linear bottleneck that limits scalability. From an optimization perspective, we identify a phenomenon we term "Policy Blindness." While gradient information in LLMs is heterogeneously distributed across layers (Wang et al., 2025), standard ZO methods employ isotropic exploration strategies that treat all parameters equally. This mismatch results in a squandering of computational resources on insensitive layers that contribute minimal learning signals.

To address these dual challenges of computational redundancy and optimization blindness, we propose **Ada**ptive **L**ayer-wise **Z**eroth-**O**rder optimization (AdaLeZO). Unlike heuristic approaches that rely on static priors, AdaLeZO formalizes the optimization process as a layer selection problem within a Multi-Armed Bandit (MAB) framework. This formulation allows the optimizer to act as a temporal filter, which integrates noisy instantaneous rewards to reveal the underlying sensitivity structure of the model. By dynamically concentrating the perturbation budget on the most critical layers, AdaLeZO achieves sparse and efficient updates. To ensure theoretical rigor, we design a gradient estimator using Inverse Probability Weighting (IPW) with replacement. This estimator ensures unbiasedness with respect to the full-parameter gradient while significantly reducing the estimation variance through importance sampling. The overall workflow of AdaLeZO is depicted in Figure 1.

We empirically validate the effectiveness of our framework on models ranging from 6.7B to 30B parameters, including the LLaMA-3.1. The results demonstrate that AdaLeZO delivers substantial wall-clock speedups of 1.7× to 3.0× over baseline methods while maintaining or surpassing competitive accuracy. Furthermore, AdaLeZO exhibits strong versatility as a plug-and-play accelerator that consistently improves the performance of advanced ZO variants, such as LoZO (Chen et al., 2025) and HiZOO (Zhao et al., 2025). In summary, our main contributions are as follows:

1. We identify the linear cost of perturbation and update operations as a major bottleneck in the ZO optimizer and reveal the structural mismatch between uniform ZO exploration and the intrinsic layer-wise sparsity of gradients.

2. We introduce AdaLeZO, a novel framework that leverages a multi-armed bandit strategy to adaptively allocate perturbations, enabling efficient sparse optimization with theoretical unbiasedness guarantees.

3. Extensive experiments confirm that AdaLeZO provides significant wall-clock acceleration and universal compatibility with existing ZO algorithms, offering a scalable solution for memory-constrained LLM fine-tuning.

2 Related Work

This section surveys the evolution of memory-efficient fine-tuning, focusing on the transition from parameter-efficient methods to ZO optimization and the integration of adaptive mechanisms.

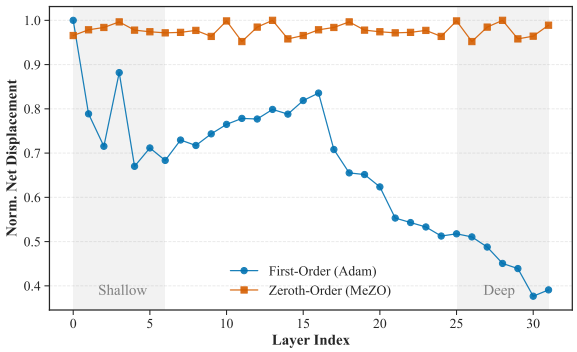
Zeroth-Order Optimization for LLMs. While Parameter-Efficient Fine-Tuning methods like LoRA (Hu et al., 2022) and Prefix-Tuning (Li and Liang, 2021) reduce trainable parameters, they remain constrained by the "memory wall" due to the storage of intermediate activations for backpropagation (Dettmers et al., 2023; Liu et al., 2022). ZO optimization has emerged as a powerful alternative, enabling LLM fine-tuning with inference-level memory footprints by estimating gradients via forward pass differences (Malladi et al., 2023; Zhang et al., 2024; Chen et al., 2024). However, standard ZO methods (e.g., MeZO) suffer from the "curse of dimensionality," where gradient estimation variance scales linearly with parameter size, leading to slow convergence (Liu et al., 2020). Although variants like Zo-Adamu (Jiang et al., 2024) and MeZO-SVRG (Gautam et al., 2024) introduce momentum or variance reduction to smooth the optimization trajectory, they largely overlook the linear wall-clock latency imposed by full-parameter perturbations. Recently, QuZO (Zhou et al., 2025) and MaZO (Zhang et al., 2025) have sought to further accelerate training via quantization, multi-task masking, or kernel transformations. However, these methods primarily optimize the temporal dimension or numerical precision. In contrast, AdaLeZO is orthogonally designed to optimize the *spatial dimension* by sparsifying $\mathcal{O}(d)$ memory reads and writes, acting as a plug-and-play module that can synergize seamlessly with these advancements for simultaneous spatial-temporal acceleration.

Subspace Exploration and Structural Priors. To mitigate the high variance of ZO estimates, a growing body of work exploits the intrinsic low-dimensional structure of LLMs. Methods such as LoZO (Chen et al., 2025), SubZero (Yu et al., 2025), and TeZO (Sun et al., 2025) constrain perturbations to low-rank matrices or tensor decompositions, effectively reducing the search space. Others, including HiZOO (Zhao et al., 2025) and LOREN (Seung et al., 2025), leverage approximate second-order information or gradient priors to guide update directions. Despite the theoretical appeal, these approaches often rely on static structural assumptions (e.g., fixed rank) or incur expensive auxiliary computations that negate wall-clock speedups. Unlike these, which impose rigid constraints, AdaLeZO focuses on *dynamic structure discovery*, autonomously identifying sensitive layers during training without pre-defined priors.

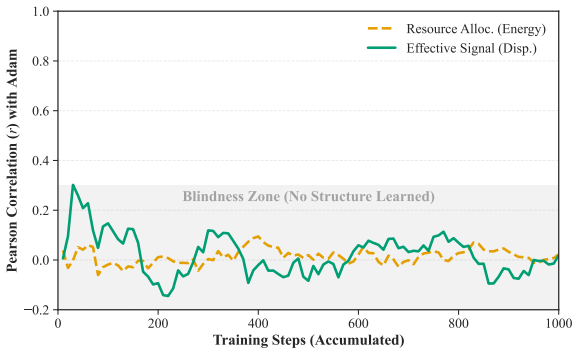
Adaptive Resource Allocation. Dynamic sparsity has proven effective in post-training pruning (Frantar and Alistarh, 2023; Sun et al., 2024), where redundant weights are removed based on activation or Hessian sensitivity. In the context of First-Order (FO) fine-tuning, layer-wise and module-wise importance sampling techniques, such as LISA (Pan et al., 2024) and MISA (Liu et al., 2025), have demonstrated substantial efficiency gains. However, these methods fundamentally rely on exact FO gradients or intermediate activation caching to evaluate structural sensitivity. Under strict ZO memory constraints, such FO priors are physically unavailable. In parallel, dynamic sampling and MAB frameworks have been successfully applied to sequential decision-making tasks in NLP, such as data selection and hyperparameter tuning (Bouneffouf and Feraud, 2025; Lin and Wang, 2023; Ceritli et al., 2024; Rao et al., 2026). AdaLeZO bridges these domains by integrating MAB into continuous ZO optimization. Crucially, it executes adaptive layer-wise importance sampling *without* FO priors, relying solely on highly noisy, forward-pass scalar loss feedback. By treating layer selection as a bandit problem, AdaLeZO realizes *adaptive update sparsity*: it dynamically concentrates the perturbation budget on the most sensitive layers. This design resolves the "blindness" of uniform ZO exploration, ensuring theoretical unbiasedness while significantly reducing computational overhead.

3 Rethinking Zeroth-Order Optimization

Despite the memory efficiency of ZO optimization, its widespread adoption is hindered by slow convergence and high variance. To elucidate the root causes of the limitations, we dissect ZO algorithms through the lenses of optimization dynamics, sys-



(a) Layer-wise Net Displacement



(b) Pearson Correlation between ZO and FO Gradient Norm

Figure 2: **Empirical demonstration of Policy Blindness.** We contrast the optimization dynamics of MeZO against those of Adam on OPT-6.7B. **(a) Layer-wise Net Displacement.** While the Adam exhibits distinct layer-wise heterogeneity by prioritizing updates on shallow layers, MeZO maintains a uniform update profile. This indicates that standard ZO methods squander the computational budget on insensitive parameters. **(b) Correlation Evolution.** The Pearson correlation between MeZO's cumulative updates and the Oracle gradient norm remains consistently low ($r < 0.2$). This confirms that isotropic perturbation fails to recover the intrinsic sensitivity structure of the model, resulting in a "blind' random walk.

tem latency, and signal fidelity. These empirical insights provide the motivational foundation for AdaLeZO.

3.1 Optimization Dynamics: Policy Blindness

A fundamental inefficiency in current ZO methods arises from a structural mismatch between the uniform exploration strategy and the heterogeneous sensitivity of LLM layers. We term this phenomenon *Policy Blindness*. To quantify this, we compare the parameter evolution of a FO oracle (Adam (Kingma and Ba, 2015)) against MeZO using *Layer-wise Net Displacement* ($||\sum \Delta\theta||_2$) as a proxy for effective learning signals. As illustrated in Figure 2, the FO oracle exhibits distinct layer-wise heterogeneity where shallow and middle layers accumulate significant updates while deeper layers remain relatively static (Clark et al., 2019; Aghajanyan et al., 2021). In stark contrast, MeZO employs an isotropic perturbation strategy that distributes the computational budget uniformly across the entire parameter space. This homogeneous allocation squanders resources on insensitive parameters that contribute negligibly to loss reduction, thereby injecting excessively high-dimensional noise that impedes convergence.

3.2 System Efficiency: Linear Bottleneck

Beyond optimization inefficiency, full-parameter perturbations introduce a critical system bottleneck. Runtime profiling on an OPT-6.7B model, as shown in Figure 3, reveals that perturbation generation and parameter updates account for approximately 46% of the total training time per step. This cost scales linearly with the parameter dimension d, rendering it a dominant factor for billion-scale models. Furthermore, classical ZO theory indicates that gradient estimation variance also scales with d (Nesterov and Spokoiny, 2017). These observations suggest that restricting perturbations to a sparse subset of sensitive layers can simultaneously eliminate this linear wall-clock overhead and reduce estimation variance, provided that the active layers are correctly identified.

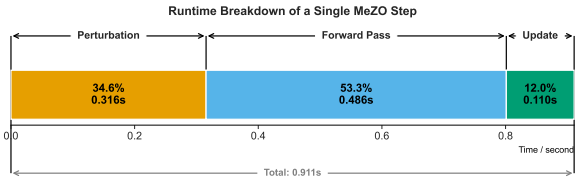


Figure 3: Time breakdown of a single training step in MeZO. The operations for perturbation generation and parameter updates constitute approximately 46% of the total step time, comparable to the forward pass time.

3.3 Signal Fidelity: Validity of Feedback

A prerequisite for adaptive layer selection is the existence of a reliable signal within the noisy ZO estimates to guide the search. We investigate whether the magnitude of the ZO gradient estimate $|\hat{g}|$ serves as a valid proxy for the true gradient norm $\|\nabla\mathcal{L}\|_F$. Our fidelity analysis in Figure 9 yields two key findings. First, at the *micro-level*, despite

the high variance intrinsic to random projection, we observe a significant positive Spearman correlation ($\rho \approx 0.48$) between the ZO estimate and the ground-truth gradient norm. Second, at the *macro-level*, binned analysis reveals a strictly monotonic relationship between the expected ZO magnitude and the true gradient norm. This statistical consistency implies that while individual ZO samples are noisy, their expectation faithfully preserves the relative ranking of layer importance. Consequently, sequential decision algorithms such as Multi-Armed Bandits can effectively exploit this property to recover the true sensitivity structure through temporal aggregation.

4 Methodology

In this section, we first revisit the standard paradigm of ZO optimization for fine-tuning LLMs and analyze its inherent computational bottlenecks. Subsequently, we propose the AdaLeZO framework. We formulate layer-wise selection as a non-stationary MAB problem and introduce a sparse gradient estimator based on IPW with replacement, achieving efficient, sparse, and low-variance optimization.

4.1 Preliminaries: The Linear Bottleneck of ZO Optimization

Consider an LLM parameterized by $\theta \in \mathbb{R}^d$. Our objective is to optimize the loss function $\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{D}}[f(\theta; x, y)]$. To circumvent the prohibitive memory cost of storing intermediate activations required by backpropagation, standard ZO methods, such as MeZO (Malladi et al., 2023), employ the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm to estimate gradients. At step t, the algorithm generates a random perturbation vector $z_t \sim \mathcal{N}(0, I_d)$ drawn from a standard normal distribution and computes the gradient estimate via two forward passes:

$$\hat{g}_t^{\text{ZO}} = \frac{\mathcal{L}(\theta_t + \mu z_t) - \mathcal{L}(\theta_t - \mu z_t)}{2\mu} z_t, \quad (1)$$

where μ denotes the smoothing parameter (perturbation radius). The parameters are updated following $\theta_{t+1} = \theta_t - \eta \hat{g}_t^{\text{ZO}}$.

Although MeZO successfully eliminates the memory bottleneck, Equation (1) reveals a fundamental flaw regarding computational efficiency: z_t is **dense**. Consequently, every optimization step necessitates sampling, adding, and updating all d parameters. As analyzed in Section 3.2, in the context of billion-scale models ($d \geq 7$B), this full-parameter operation incurs a linear time complexity of $O(d)$, accounting for over 40% of the total training time. Furthermore, applying uniform perturbation across the entire high-dimensional parameter space introduces significant estimation variance, known as the "curse of dimensionality" (Liu et al., 2020; Nesterov and Spokoiny, 2017), which severely hampers convergence.

4.2 Adaptive Layer Selection via Multi-Armed Bandit

To mitigate the aforementioned computational redundancy and high variance, AdaLeZO leverages the layer-wise heterogeneity of LLM gradients by modeling the selection of trainable layers as a Multi-Armed Bandit (MAB) problem. We partition the model parameters into L groups (layers), denoted as $\{\theta^{(1)}, \ldots, \theta^{(L)}\}$. Our goal is to dynamically learn a sampling policy π_t that allocates the limited perturbation budget to the layers most sensitive to the loss function.

Reward Definition. We require a reward signal to quantify the contribution of a specific layer to the optimization process. We define the immediate reward R_t at step t as the magnitude of the estimated scalar gradient. Intuitively, if a perturbation induces a significant change in the loss, it indicates that the optimization direction lies in a steep region of the loss landscape, thereby offering higher optimization value:

$$R_t = \left| \frac{\mathcal{L}(\theta_t + \mu z_t) - \mathcal{L}(\theta_t - \mu z_t)}{2\mu} \right|. \quad (2)$$

This scalar serves as a proxy for the effectiveness of the current step. As analyzed in Appendix F, the optimal sampling probability is proportional to the gradient norm, i.e., $p_t(l) \propto \|\nabla^{(l)} \mathcal{L}\|$, validating R_t as an ideal proxy signal.

Value Estimation (EMA). Since the sensitivity of layers varies dynamically during training (i.e., the environment is non-stationary), we employ an Exponential Moving Average (EMA) to maintain the value estimate $Q_t(l)$ for each layer. For every layer l in the selected active set $\mathcal{I}_t$ at step t, the value is updated as:

$$Q_{t+1}(l) = (1 - \alpha) Q_t(l) + \alpha R_t, \quad (3)$$

where α is the learning rate factor. For unselected layers, Q remains unchanged. This design allows

the algorithm to smooth out historical noise while rapidly adapting to the distribution shift in layer importance.

Policy with Exploration-Exploitation. To balance the exploitation of highly sensitive layers with the exploration of under-sampled ones, we compute the sampling probability distribution p_t based on the current Q_t values. For the l-th layer:

$$p_t(l) = (1-\gamma) \cdot \underbrace{\text{Softmax}(Q_t(l)/\tau)}_{\text{Exploitation}} + \gamma \cdot \underbrace{\frac{1}{L}}_{\text{Exploration}}, \tag{4}$$

where τ is the temperature coefficient controlling distribution smoothness, and $\gamma \in [0, 1]$ is the mixing coefficient. The inclusion of the uniform distribution term $\frac{1}{L}$ is critical; it guarantees a non-zero lower bound probability $p_t(l) \geq \gamma/L$, preventing the permanent "starvation" of layers, which is a necessary condition for the unbiasedness of the subsequent estimator.

4.3 Sparse Gradient Estimation with IPW

Based on the probability distribution p_t, we design a sparse gradient estimator utilizing Sampling with Replacement and IPW.

Sampling Mechanism. Unlike previous methods that employ sampling without replacement (Top-k), AdaLeZO performs $K = \max(1, \lfloor \rho L \rfloor)$ independent draws **with replacement** based on p_t at step t, where $\rho \in (0, 1]$ is the sampling ratio. Let $n_{t,l}$ denote the number of times layer l is selected (multiplicity). If $n_{t,l} > 0$, the layer is marked as active, forming the set $\mathcal{I}_t = \{l \mid n_{t,l} > 0\}$.

Sparse Perturbation. We generate Gaussian noise only for the active layers to construct a sparse perturbation vector $\tilde{z}_t$:

$$\tilde{z}_t^{(l)} = \begin{cases} \mathcal{N}(0, I_{d_l}) & \text{if } l \in \mathcal{I}_t, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Since $|\mathcal{I}_t| \leq K \ll L$, this operation significantly reduces the overhead of perturbation generation and parameter updates from $O(d)$ to $O(\rho d)$.

Count-Aware IPW Estimator. Directly using sparse perturbation introduces bias. To address this, we apply importance-sampling reweighting, incorporating the counting property of sampling with replacement, the AdaLeZO gradient estimate for layer l is defined as:

$$\hat{g}_t^{\text{Ada},(l)} = \hat{g}_{\text{scalar}} \cdot w_{t,l} \cdot n_{t,l} \cdot \tilde{z}_t^{(l)}, \tag{6}$$

where the projected gradient scalar is $\hat{g}_{\text{scalar}} \triangleq \frac{\mathcal{L}(\theta_t + \mu\tilde{z}_t) - \mathcal{L}(\theta_t - \mu\tilde{z}_t)}{2\mu}$, and the IPW weight is $w_{t,l} \triangleq \min\left(\frac{1}{K p_t(l)}, C_{\text{clip}}\right)$. In this formulation, the multiplicity $n_{t,l}$ leverages the statistical nature of sampling with replacement, ensuring that important layers selected multiple times ($n_{t,l} > 1$) receive larger update steps to automatically intensify optimization on sensitive parameters. Simultaneously, the IPW weight $w_{t,l}$ functions as a clipped inverse probability weight which, as proved in Appendix F, converges to an unbiased estimator of the full-parameter Gaussian smoothed gradient, i.e., $\mathbb{E}[\hat{g}_t^{\text{Ada}}] = \nabla \mathcal{L}_\mu(\theta_t)$, as the clipping threshold $C_{\text{clip}} \to \infty$. Furthermore, regarding variance reduction, while standard IPW ensures unbiasedness, extremely small sampling probabilities $p_t(l)$ can induce numerical instability; thus, the clipping threshold C_{clip} is introduced to establish a bias-variance trade-off that significantly reduces estimation variance in practical scenarios.

The final parameter update is performed strictly on active layers: $\theta_{t+1}^{(l)} = \theta_t^{(l)} - \eta \hat{g}_t^{\text{Ada},(l)}$. The complete algorithmic procedure is detailed in Algorithm 1. We further provide a rigorous theoretical proof in Appendix F demonstrating that AdaLeZO maintains a convergence rate of $O(1/\sqrt{T})$.

5 Experiments

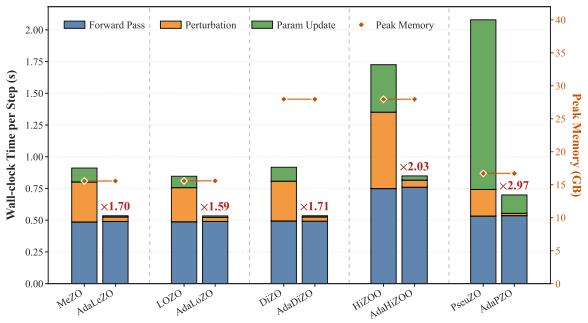


Figure 4: Breakdown of wall-clock time per training step and peak memory consumption across different ZO optimizations. The stacked bars represent the time cost of forward pass, perturbation generation, and parameter update, while the orange diamonds indicate peak memory usage. Our proposed Ada- methods significantly compress the overhead of perturbation and updates.

5.1 Experimental Setup

We evaluate AdaLeZO on LLaMA-2-7B (Touvron et al., 2023), LLaMA-3.1-8B (Grattafiori et al., 2024), and OPT 6.7B-30B (Zhang et al., 2022) models across 11 downstream tasks. We compare

Table 1: **Main Results on LLaMA Series.** We compare AdaLeZO against the Zero-shot baseline and MeZO (Full-parameter ZO). The results are averaged over 3 random seeds. The best performance between ZO methods is marked in **bold**.

Method	Classification							Multiple Choice		Generation		AVG.
	SST-2	RTE	CB	BoolQ	WSC	WIC	MultiRC	Copa	ReCoRD	SQuAD	DROP	
LLaMA-2-7B												
Zero-shot	58.14	63.18	32.14	70.60	36.54	50.16	45.50	78.00	80.70	55.47	20.20	53.69
MeZO	93.92±0.90	63.66±2.08	68.45±2.73	**78.60**±0.52	**60.58**±3.33	61.91±3.70	**71.77**±3.44	**84.33**±1.53	81.03±0.70	**88.64**±0.63	40.95±0.69	72.17±1.84
AdaLeZO	**94.00**±0.13	**66.19**±2.71	**69.05**±2.73	76.30±1.85	60.26±3.89	**62.91**±2.51	70.43±2.97	84.00±1.00	**81.40**±0.82	88.49±0.30	**41.98**±1.27	**72.27**±1.83
LLaMA-3.1-8B												
Zero-shot	59.75	46.57	44.64	76.30	50.78	59.62	62.30	64.88	83.70	85.00	28.46	60.18
MeZO	**93.81**±0.41	**75.45**±2.50	70.24±1.03	80.73±0.80	**62.18**±1.11	58.03±1.94	74.87±1.35	89.67±0.58	83.90±1.00	89.37±1.26	60.78±1.23	76.28±1.20
AdaLeZO	92.93±0.92	71.24±5.26	**72.02**±2.73	**81.20**±0.96	61.22±2.22	**59.51**±0.80	**80.17**±0.76	**90.33**±0.58	**84.70**±1.05	**89.74**±0.78	**61.87**±0.31	**76.81**±1.49

Table 2: **Universal Effectiveness on OPT-6.7b.** We report the accuracy (%) and standard deviation across three random seeds. We compare various ZO methods with their adaptive counterparts powered by our AdaLeZO framework. The best result in each pair (e.g., MeZO vs. AdaLeZO) is marked in **bold**.

Method	Classification							Multiple Choice		Generation		AVG.
	SST-2	RTE	CB	BoolQ	WSC	WIC	MultiRC	Copa	ReCoRD	SQuAD	DROP	
Zero-shot	61.24	54.87	50.00	63.10	37.50	51.25	44.50	82.00	76.00	36.48	17.70	52.24
ICL	84.29	65.70	57.14	70.40	51.92	53.61	50.20	81.00	76.90	74.30	27.75	63.02
FT (Upper Bound)	92.78	78.34	94.64	73.90	59.62	51.25	77.50	80.00	75.30	85.28	28.53	72.47
MeZO	93.54±0.48	65.46±2.08	69.05±1.03	67.13±0.31	**56.09**±5.47	58.78±1.10	**66.10**±2.51	**81.33**±1.53	78.03±0.60	**81.52**±1.43	28.79±1.04	67.80±1.60
AdaLeZO	**93.58**±0.23	**67.39**±1.50	**70.24**±2.73	**67.93**±0.49	55.45±4.74	**59.61**±1.16	63.13±1.67	80.33±2.52	**79.07**±0.49	80.65±0.23	**29.19**±0.99	**67.87**±1.52
LOZO	**93.69**±0.61	**67.51**±2.73	69.64±1.79	**69.53**±0.38	50.64±11.23	59.30±3.76	59.87±1.50	79.00±1.00	78.83±0.55	**81.65**±0.19	**29.49**±2.51	67.20±2.39
AdaLoZO	93.27±0.07	66.19±1.27	**72.02**±2.73	67.33±0.45	**56.09**±2.94	**62.07**±0.87	**60.30**±0.30	**80.67**±0.58	**79.17**±0.55	78.03±1.59	28.76±1.37	**67.63**±1.16
DiZO	92.93±0.13	**67.03**±1.63	**69.64**±3.09	**68.77**±2.79	**60.90**±4.54	**62.00**±1.31	62.03±1.07	78.33±2.31	78.13±0.21	**80.38**±1.34	25.93±1.65	67.83±1.83
AdaDiZO	**93.04**±0.63	66.79±0.63	69.05±1.03	68.20±0.46	59.30±3.09	61.02±1.28	**62.97**±0.75	**82.33**±2.52	**78.57**±0.21	79.59±1.82	**29.24**±1.45	**68.19**±1.26
HiZOO	**93.43**±0.48	**65.46**±2.40	69.05±1.03	**68.03**±1.46	56.41±5.80	59.25±1.10	**65.57**±2.97	**82.00**±2.00	78.13±0.29	**81.96**±1.35	27.81±1.18	**67.92**±1.82
AdaHiZOO	92.85±0.52	64.86±1.85	**71.43**±1.79	67.17±1.27	**56.73**±5.85	**60.24**±1.31	64.23±0.15	80.33±0.58	**78.97**±0.74	80.12±1.88	**28.75**±0.75	67.79±1.52
PseuZO	**93.77**±0.07	**65.22**±2.76	**71.43**±1.79	67.30±0.62	**56.09**±11.27	**61.55**±1.19	59.67±1.55	79.00±1.00	**78.47**±0.50	78.60±1.28	26.08±2.30	67.02±2.21
AdaPZO	93.00±0.40	64.50±1.37	**71.43**±0.00	**67.67**±2.14	55.13±5.80	61.34±0.39	**60.23**±0.50	**80.67**±1.15	**78.47**±0.64	**78.67**±0.96	**28.57**±0.90	**67.24**±1.30

against MeZO (Malladi et al., 2023), LoZO (Chen et al., 2025), DiZO (Tan et al., 2025), HiZOO (Zhao et al., 2025), and PseuZO (Yue et al., 2025). Detailed hyperparameters and baselines are listed in Appendix A.

5.2 Efficiency Analysis: Breaking the Linear Barrier

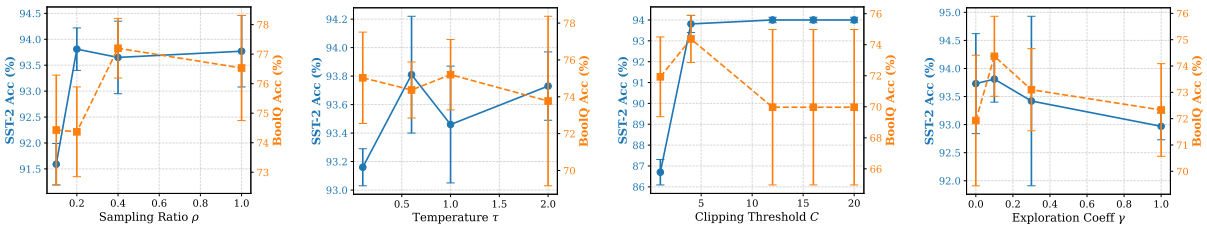
We analyze the core bottleneck in ZO optimization: the linear complexity of perturbation operations. Figure 4 presents a comparative visualization of wall-clock latency and memory usage for AdaLeZO versus standard ZO baselines.

Wall-Clock Acceleration. Standard MeZO suffers from structural inefficiency, with perturbation generation and parameter updates accounting for nearly half of each training step. By restricting these operations to a sparse set of active layers, AdaLeZO effectively removes this bottleneck. To rigorously ensure fairness, we evaluate the absolute hardware throughput on a single NVIDIA A100 (40GB) GPU in BF16 precision. For OPT-6.7B (batch size 16, sequence length 256), AdaLeZO achieves a throughput of 7,728 tokens/sec, significantly surpassing MeZO's 4,501 tokens/sec. This translates to a genuine $1.7\times$ hardware-level speedup, confirming that the MAB bookkeeping overhead is negligible ($\ll 1\%$ of step time). Furthermore, when applied to PseuZO, the speedup reaches almost $3.0\times$ by circumventing costly projection steps. Detailed results in Table 7 show that perturbation overhead is reduced to less than 10% of step time.

Memory Neutrality. Notably, AdaLeZO preserves the peak memory footprint of standard ZO methods, as the orange diamonds shown in Figure 4. In contrast to FO approaches that require substantial amounts of optimizer state, AdaLeZO operates entirely within inference-level memory constraints, ensuring compatibility with consumer hardware. As shown in Table 8, the efficiency improvements are consistent across sequence lengths, achieving up to $5.1\times$ speedup on short-sequence tasks where perturbation overhead is most significant.

5.3 Main Results on LLaMA Series

Table 1 presents the performance of MeZO and AdaLeZO on LLaMA-2-7B and LLaMA-3.1-8B



(a) Effect of Sampling Ratio ρ (b) Effect of Temperature τ (c) Effect of IPW Clipping C (d) Effect of Exploration γ

Figure 5: **Ablation Studies.** We analyze the impact of four key hyperparameters on SST-2 (blue, left axis) and BoolQ (orange, right axis) performance. Error bars denote standard deviation across 3 seeds. Detail in Table 4.

models. AdaLeZO demonstrates consistent superiority over the dense baseline.

Superiority over Dense Updates. AdaLeZO achieves highly competitive performance against, and often surpasses, the full-parameter MeZO baseline. On challenging reasoning benchmarks such as DROP and SQuAD, AdaLeZO delivers notable improvements, including a gain of 1.03% on DROP with LLaMA-2. These results challenge the notion that updating more parameters necessarily leads to better outcomes, and demonstrate that adaptive sparse updates can effectively suppress the detrimental noise associated with high-dimensional dense perturbations.

Stability and Robustness. ZO optimization is often hindered by high variance. AdaLeZO achieves lower standard deviations compared to MeZO, indicating greater stability. This improvement stems from the count-aware IPW estimator, which, as detailed in Section 4, reduces estimation variance by focusing updates on frequently sampled, high-sensitivity layers.

5.4 Universality across ZO Optimizers

A defining feature of AdaLeZO is its orthogonality to existing ZO enhancements. We integrate AdaLeZO as a plug-and-play module into four representative baselines. As shown in Table 2, the AdaLeZO-enhanced variants (denoted as Ada-) consistently improve or maintain the accuracy of their base methods while providing the significant speedups discussed in Section 5.2. For instance, AdaHiZOO not only accelerates HiZOO by $2.0\times$ but also improves average accuracy by 0.15% on OPT-6.7B. This universality confirms that adaptive layer-wise sparsity captures a fundamental property of the loss landscape that is complementary to subspace constraints (LoZO) or gradient priors (HiZOO). Additionally, we demonstrate the scalability of AdaLeZO on OPT-13B and OPT-30B models in Table 6, where it continues to outperform MeZO, validating its effectiveness for large-scale fine-tuning.

5.5 Ablation Study

We conduct comprehensive ablation studies with the LLaMA-2-7b model to evaluate the contributions of each AdaLeZO component and hyperparameter sensitivity. Two representative tasks are selected: SST-2 (robust to noise) and BoolQ (sensitive to gradient estimation errors). Key results are shown in Figure 5.

Necessity of Sparse Update Figure 5(a) shows the effect of sampling ratio ρ. Performance improves markedly as ρ increases from 0.1 to 0.2, suggesting that excessive sparsity may discard crucial gradient information. However, further increasing ρ to 0.4 or 1.0 (full-parameter MeZO) yields diminishing or negative returns. This supports our hypothesis: **full-parameter perturbation is often suboptimal for zeroth-order optimization**. Given the ZO error bound $O(d/\sqrt{T})$, estimation variance scales with dimension d. By reducing the update space to ρd, AdaLeZO lowers variance and introduces implicit regularization, outperforming dense baselines. Furthermore, to definitively isolate the optimization benefits of our MAB-driven policy from the wall-clock speedups of raw sparsity, we demonstrate in Appendix C that AdaLeZO consistently outperforms a uniform Random Sparse baseline (by 2.12% on average), proving the necessity of adaptive layer selection.

Variance Reduction is Critical Figure 5(c) highlights the sensitivity to the IPW clipping threshold C. At $C = 1$ (unweighted sparse updates), performance drops due to sampling bias. Conversely, a large C (≥ 12) increases estimator variance, causing accuracy on BoolQ to collapse (from 74.37% to 69.97%), as extreme weights $1/(Kp_l)$ amplify

gradient variance. Setting $C = 4$ achieves an optimal balance between **bias** and **variance**, which is crucial for high-dimensional ZO optimization.

Balancing Exploration and Exploitation Figure 5(b) and (d) analyze the bandit strategy. A small temperature ($\tau = 0.1$, nearly greedy) leads to premature convergence, while a large τ ($= 2.0$, nearly uniform) fails to exploit gradient sensitivity; both underperform compared to $\tau = 0.6$. The exploration coefficient γ is also essential: adding slight uniform exploration ($\gamma = 0.1$) boosts BoolQ performance by 2.4% over pure Softmax ($\gamma = 0$), confirming that preventing "layer starvation" is vital for estimator coverage and convergence in non-stationary bandit settings.

6 Conclusion

In this work, we identify the linear computational cost of perturbation and parameter updates as a fundamental bottleneck in zeroth-order (ZO) large language model (LLM) fine-tuning. To address this limitation, we introduce AdaLeZO, a novel framework that integrates multi-armed bandit (MAB)-driven adaptive layer selection with a count-aware inverse probability weighting (IPW) estimator, enabling efficient, sparse, and unbiased gradient estimation. By dynamically concentrating perturbations on the most sensitive layers, AdaLeZO effectively mitigates the curse of dimensionality and offers theoretical guarantees for convergence. Extensive experiments on both LLaMA and OPT models demonstrate that AdaLeZO achieves $1.7\times$ to $5.1\times$ wall-clock speedup over state-of-the-art baselines, without any loss in accuracy. Furthermore, the plug-and-play design of AdaLeZO allows for seamless integration with existing ZO optimizers, providing a unified and scalable solution for memory-efficient and rapid LLM fine-tuning.

Limitations

Although AdaLeZO delivers substantial improvements in wall-clock efficiency and convergence stability, several inherent limitations of the Zeroth-Order optimization paradigm remain.

Performance Gap with First-Order Methods. AdaLeZO consistently surpasses standard zeroth-order baselines such as MeZO and narrows the gap with full fine-tuning. However, its performance does not fully reach that of first-order methods across all tasks. For example, in challenging reasoning benchmarks such as DROP, the stochastic nature of gradient estimation, even when enhanced by variance reduction techniques, results in a precision trade-off relative to backpropagation. Addressing this gap continues to be a fundamental challenge for the zeroth-order optimization community.

Scope of Application. The present evaluation is limited to supervised fine-tuning of large language models within the field of natural language processing. The effectiveness of adaptive layer-wise sampling for other areas, including multimodal large language models and reinforcement learning from human feedback, has not yet been empirically validated. Investigating these extensions represents an important direction for future research.

Acknowledgements

This work was supported by the National Key R&D Projects under Grant 2024YFC3307100; the National Natural Science Foundation of China under Grants 62076101, 62576364, and 62306118; the Guangdong Basic and Applied Basic Research Foundation under Grants 2024B1515020082, 2023A1515010007, 2026B1515020071, and 2026A1515010725; the Guangdong Provincial Key Laboratory of Human Digital Twin under Grant 2022B1212010004; the Shenzhen Basic Research Project (Natural Science Foundation) Basic Research Key Project under Grant JCYJ20241202124430041; the Fundamental Research Funds for the Central Universities under Grant 2025ZYGXZR054; the TCL Young Scholars Program; and the 2024 Tencent AI Lab Rhino-Bird Focused Research Program.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. 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.

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, and 29 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*, 7(8):1.

Djallel Bouneffouf and Raphael Feraud. 2025. Multi-armed bandits meet large language models. *arXiv preprint arXiv:2505.13355*.

Aichen Cai, Anmeng Zhang, Anyu Li, Bo Zhang, Bohua Cai, Chang Li, Changjian Jiang, Changkai Lu, Chao Xue, Chaocai Liang, Cheng Zhang, Dongkai Liu, Fei Wang, Guoqiang Huang, Haijian Ke, Han Lin, Hao Wang, Ji Miao, Jiacheng Zhang, and 50 others. 2026. Joyai-llm flash: Advancing mid-scale llms with token efficiency. *arXiv preprint arXiv:2604.03044*.

Taha Ceritli, Savas Ozkan, Jeongwon Min, Eunchung Noh, Cho Jung Min, and Mete Ozay. 2024. A study of parameter efficient fine-tuning by learning to efficiently fine-tune. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15819–15836.

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

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

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

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286. Association for Computational Linguistics.

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

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, and 69 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

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

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR.

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

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

Shuoran Jiang, Qingcai Chen, Youcheng Pan, Yang Xiang, Yukang Lin, Xiangping Wu, Chuanyi Liu, and Xiaobao Song. 2024. 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, pages 18363–18371.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, pages 1–13.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Xiang Lisa Li and Percy Liang. 2021. 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)*, pages 4582–4597.

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and Zhiyu Li. 2024. Controllable text generation for large language models: A survey. *arXiv preprint arXiv:2408.12599*.

Yiqi Lin and Ru Wang. 2023. Bandit-nas: Bandit sampling method for neural architecture search. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

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

Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. 2020. 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.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Yuxi Liu, Renjia Deng, Yutong He, Xue Wang, Tao Yao, and Kun Yuan. 2025. MISA: Memory-efficient LLMs optimization with module-wise importance sampling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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

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

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. 2024. LISA: Layerwise importance sampling for memory-efficient large language model fine-tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. 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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. 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.

Jun Rao, Zepeng Lin, Xuebo Liu, Xiaopeng Ke, Lian Lian, Dong Jin, Shengjun Cheng, Jun Yu, and Min Zhang. 2025. APT: Improving specialist LLM performance with weakness case acquisition and iterative preference training. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20958–20980, Vienna, Austria. Association for Computational Linguistics.

Jun Rao, Xuebo Liu, Hexuan Deng, Zepeng Lin, Zixiong Yu, Jiansheng Wei, Xiaojun Meng, and Min Zhang. 2026. Dynamic sampling that adapts: Self-aware iterative data persistent optimization for mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*.

Jun Rao, Xuebo Liu, Lian Lian, Shengjun Cheng, Yunjie Liao, and Min Zhang. 2024. CommonIT: Commonality-aware instruction tuning for large language models via data partitions. In *EMNLP*, pages 10064–10083, Miami, Florida, USA. Association for Computational Linguistics.

Jun Rao, Fei Wang, Liang Ding, Shuhan Qi, Yibing Zhan, Weifeng Liu, and Dacheng Tao. 2022. Where does the performance improvement come from - a reproducibility concern about image-text retrieval. In *SIGIR*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Hyunseok Seung, Jaewoo Lee, and Hyunsuk Ko. 2025. Low-rank curvature for zeroth-order optimization in llm fine-tuning. *arXiv preprint arXiv:2511.07971*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and

Christopher Potts. 2013. 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.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*.

Yan Sun, Tiansheng Huang, Liang Ding, Li Shen, and Dacheng Tao. 2025. Tezo: Empowering the low-rankness on the temporal dimension in the zeroth-order optimization for fine-tuning llms. *arXiv preprint arXiv:2501.19057*.

Qitao Tan, Jun Liu, Zheng Zhan, Caiwen Ding, Yanzhi Wang, Xiaolong Ma, Jaewoo Lee, Jin Lu, and Geng Yuan. 2025. 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*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Fei Wang, Li Shen, Liang Ding, Chao Xue, Ye Liu, and Changxing Ding. 2024. Simultaneous computation and memory efficient zeroth-order optimizer for fine-tuning large language models. *arXiv preprint arXiv:2410.09823*.

Fei Wang, Li Shen, Liang Ding, Chao Xue, Ye Liu, and Changxing Ding. 2025. Layer as puzzle pieces: Compressing large language models through layer concatenation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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

Pengyun Yue, Xuanlin Yang, Mingqing Xiao, and Zhouchen Lin. 2025. PseuZO: Pseudo-zeroth-order algorithm for training deep neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

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. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D. Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen. 2024. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark. In *Forty-first International Conference on Machine Learning*.

Zhen Zhang, Yifan Yang, Kai Zhen, Nathan Susanj, Athanasios Mouchtaris, Siegfried Kunzmann, and Zheng Zhang. 2025. Mazo: Masked zeroth-order optimization for multi-task fine-tuning of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18537–18554.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. Galore: Memory-efficient LLM training by gradient low-rank projection. In *Forty-first International Conference on Machine Learning*.

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

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

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781.

Table 3: **Hyperparameter Settings.** We list the common settings shared across all ZO methods and the specific hyperparameters for each baseline and AdaLeZO.

Category	Hyperparameter	Value
	Common Settings for All ZO Methods	
General	Batch Size	16
	Learning Rate	$\{1, 5, 10\} \times 10^{-7}$
	Perturbation Scale μ	1×10^{-3}
	Method-Specific Settings	
LOZO	Rank r	2
	Interval v	50
HiZOO	Estimate times n	1
	Smooth scale α	1×10^{-8}
DiZO	Projection Update Cycle	100
	Projection Iterations	10
	Smoothing Scalar	0.1
	Projection Step Size	2
	Clip Range	0.2
PseuZO	Sliding Window L	14
	Cycles / Epochs	2 / 10
	Descent Formula	$\lambda(t) = \frac{\lambda_{\max}}{1+10t}$
AdaLeZO (Ours)	Sampling Ratio ρ	0.2
	Temperature τ	0.6
	Exploration Coeff. γ	$\{0, 0.1\}$
	EMA Factor α	0.1
	Clipping Threshold C	$\{4, 16\}$
	Fine-Tuning (Reference)	
FT (Adam)	Batch Size	8
	Learning Rate	1×10^{-5}
	Scheduler	Linear

A Experiments Detial

A.1 Baseline

We conducted comparative evaluations of AdaLeZO against three baseline approaches: zero-shot, in-context learning (ICL), fine-tuning (FT), and MeZO (Malladi et al., 2023). The zero-shot approach assesses both pre-trained models without any fine-tuning, serving as a lower-bound performance baseline. Fine-tuning (FT) is referenced to indicate the performance of non-quantized models. To further investigate the generalizability of AdaLeZO, we also compared it with four state-of-the-art MeZO-based zero-order optimization methods: LoZO (Chen et al., 2025), DiZO (Tan et al., 2025), HiZOO (Zhao et al., 2025), and PseuZO (Yue et al., 2025).

A.2 Models, Dataset, and Metrics

Our experiments encompass two prominent model families: the OPT (Zhang et al., 2022) family and the LLaMA (Touvron et al., 2023; Grattafiori et al., 2024) family. To ensure comprehensive coverage of model scales, we selected four models from the OPT (Zhang et al., 2022) family: OPT-6.7B, OPT-13B, and OPT-30B. For contemporary relevance, we included LLaMA-2-7B (Touvron et al., 2023) and LLaMA-3.1-8B (Grattafiori et al., 2024) from the LLaMA family.

For downstream evaluation, we employed eleven tasks commonly used in ZO optimization literature. These include seven classification tasks from the SuperGLUE (Wang et al., 2019) benchmark: SST-2 (Socher et al., 2013), RTE (Bentivogli et al., 2009), CB (De Marneffe et al., 2019), BoolQ (Zhang et al., 2018), WIC (Pilehvar and Camacho-Collados, 2019), WSC (Levesque et al., 2012), and MultiRC (Khashabi et al., 2018); two multiple-choice tasks from SuperGLUE benchmark: Copa (Roemmele et al., 2011) and ReCoRD (Zhang et al., 2018); and two question-answering tasks: SQuAD (Rajpurkar et al., 2016) and DROP (Dua et al., 2019), which we treat as generation tasks. For classification and multiple-choice tasks, we report accuracy; for generation tasks, we report the F1 score.

A.3 Implementation

For all ZO methods, we ensured fair comparison by adopting the optimal hyperparameter settings from their original publications and conducting a grid search over learning rates. Detailed hyperparameter configurations are provided in Table 3. For AdaLeZO, we used the default parameters specified in Table 3 across all experiments, except in ablation studies where only the parameters under investigation were modified. AdaLeZO employs the same learning rate as the baseline for comparison.

During training, all models were fine-tuned for 20,000 steps with checkpoints saved every 5,000 steps. We selected the checkpoint with the lowest validation loss for final evaluation on the test set. For each task, we used 1,000 samples for training and 500 for validation; when fewer than 1,000 samples were available, we used 100 for validation and the remainder for training. All available test samples were used for evaluation. To ensure statistical reliability, we repeated each experiment with three different random seeds and report the average performance metric (Rao et al., 2022).

B Additional Ablation Studies

In this section, we provide a granular analysis of the hyperparameters governing the Multi-Armed Bandit (MAB) mechanism in AdaLeZO, specifically

Table 4: **Ablation Study on LLaMA-2-7b.** We report the average accuracy and standard deviation across 3 seeds. Default parameters are underlined. **Bold** indicates the best performance. This table validates the contribution of each component in our Bandit-based strategy.

Ablation Component (Parameter)	**SST-2**		**BoolQ**	
	Avg.	**Std.**	**Avg.**	**Std.**
Sampling Ratio ρ				
0.1	91.59	0.40	74.43	1.86
0.2	**93.81**	0.41	74.37	1.52
0.4	93.65	0.70	**77.20**	1.01
1.0 (Full)	93.77	0.69	76.53	1.78
Temperature τ				
0.1 (Greedy)	93.16	0.13	**75.03**	2.48
0.6	**93.81**	0.41	74.37	**1.52**
1.0	93.46	0.41	75.20	1.91
2.0 (Uniform)	93.73	0.24	73.77	4.60
Clipping Threshold C				
1	86.70	0.61	71.93	2.57
4	93.81	0.41	**74.37**	**1.52**
12	**94.00**	0.13	69.97	5.01
16	**94.00**	0.13	69.97	5.01
Exploration γ				
0.0 (Pure Softmax)	93.73	0.89	71.93	2.48
0.1	**93.81**	0.41	**74.37**	**1.52**
0.3	93.42	1.51	73.10	1.56
1.0 (Uniform)	92.97	0.24	72.33	1.76
EMA Factor α				
0.1	**93.81**	0.41	74.37	**1.52**
0.5	92.85	0.59	**75.10**	3.32

the Reward EMA factor α, Temperature τ, and Exploration coefficient γ. These parameters are critical for balancing the trade-off between plasticity and stability in the non-stationary optimization landscape of LLM fine-tuning. The full numerical comparisons are detailed in Table 4.

B.1 Sensitivity to Bandit Hyperparameters

Reward EMA Factor (α). The EMA factor α controls how quickly the bandit forgets historical rewards. A larger α makes the policy more responsive to recent gradient estimates but also more susceptible to instantaneous noise. Comparing the default $\alpha = 0.1$ with a more aggressive update rate $\alpha = 0.5$, we observe in Table 4 that while $\alpha = 0.5$ achieves comparable average accuracy, it introduces significant instability. Specifically, on the BoolQ dataset, the standard deviation for $\alpha = 0.5$ is more than double that of $\alpha = 0.1$ (3.32 vs. 1.52). This result corroborates that in Zeroth-Order optimization, where gradient estimates naturally possess high variance, a lower α is preferable as it effectively smooths out the noise, providing a stable signal for layer importance.

Temperature (τ). The temperature parameter τ modulates the sharpness of the Softmax distribution derived from the estimated Q-values. Our results indicate that extreme values are detrimental to performance. A low temperature ($\tau = 0.1$) approximates a greedy strategy, which risks premature convergence to suboptimal layers and leads to high performance variance (Std 2.48 on BoolQ). Conversely, a high temperature ($\tau = 2.0$) approaches uniform sampling, diluting the benefits of adaptive selection. The choice of $\tau = 0.6$ strikes an optimal balance, allowing the model to exploit sensitive layers while maintaining sufficient entropy in the sampling distribution.

Exploration Coefficient (γ). We further verify the necessity of the explicit exploration term γ/L in Equation (4). The empirical results demonstrate that a small mixing coefficient $\gamma = 0.1$ consistently outperforms the pure Softmax strategy ($\gamma = 0$), particularly on challenging tasks. This confirms that ensuring a non-zero lower bound on sampling probabilities is crucial for preventing "layer starvation", which is a scenario where potentially useful layers are permanently ignored due to early stochastic fluctuations, thereby maintaining the asymptotic validity of the estimator. **When $\gamma = 1$, the method degenerates to random search, resulting in a significant performance decline on the SST-2 and BoolQ benchmarks. This observation indirectly corroborates the effectiveness of the proposed Multi-Armed Bandit strategy.**

C Impact of the Multi-Armed Bandit Policy versus Random Sparsity

To empirically validate the necessity of the proposed MAB-driven adaptive layer selection, we must isolate the optimization benefits of our policy from the wall-clock speedups provided by raw sparsity. To this end, we introduce a **Random Sparse** baseline that selects layers uniformly at random at each step, strictly utilizing the same sparsity budget as AdaLeZO ($\rho = 0.2$).

We evaluate this baseline on the LLaMA-2-7B model across a 9-task subset of our evaluation suite. The results are summarized in Table 5.

As shown in Table 5, AdaLeZO consistently out-

Table 5: Comparison between AdaLeZO and the Random Sparse baseline on LLaMA-2-7B. Both methods operate under identical sparsity budgets ($\rho = 0.2$). The MAB policy yields a +2.12% average improvement.

Method	**SST-2**	**RTE**	**BoolQ**	**WSC**	**WiC**	**MultiRC**	**SQuAD**	**ReCoRD**	**AVG.**
Random Sparse	92.35	64.62	73.57	61.33	58.97	61.43	87.88	80.73	72.22
AdaLeZO	**94.00**	**66.19**	**76.30**	**62.51**	**60.20**	**70.43**	**88.49**	**81.40**	**74.34**
Improvement	*+1.65*	*+1.57*	*+2.73*	*+1.18*	*+1.23*	*+9.00*	*+0.61*	*+0.67*	***+2.12***

performs uniform random selection by 2.12% on average. The performance gap is particularly pronounced on complex reasoning and comprehension tasks, with significant gains of up to 9.00% on MultiRC and 2.73% on BoolQ.

This rigorously confirms that the MAB policy is strictly necessary. While raw sparsity alone provides the computational acceleration, uniformly discarding layers leads to a severe loss of critical gradient signals. AdaLeZO effectively identifies and focuses updates on the most sensitive parameters, preventing degradation and maintaining robust accuracy under high sparsity constraints.

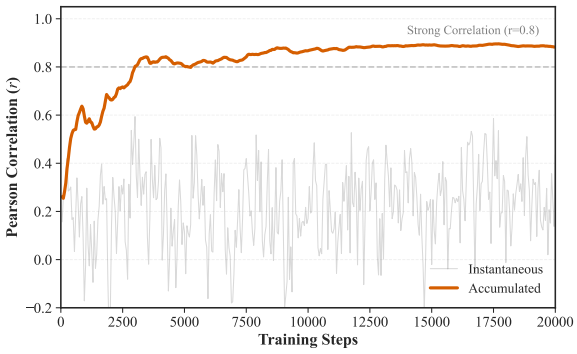


Figure 6: The Pearson correlation between the layer sampling probabilities assigned by AdaLeZO and the gradient norms computed by Adam. Instantaneous correlation exhibits substantial fluctuations, reflecting the high variance inherent in ZO estimates. In contrast, accumulated statistics converge steadily to $r \approx 0.88$, demonstrating that AdaLeZO effectively recovers true layer sensitivity through temporal aggregation.

D Mechanism Analysis

Analyzing the Structural Learning Capability of AdaLeZO. A central question arises: *Given only noisy Zeroth-Order (ZO) scalar feedback, does AdaLeZO genuinely "learn" the hierarchical structure of the model, or is it merely performing a random walk?*

To address this, we tracked the Pearson correlation coefficient r between the layer sampling probability distribution π_t of AdaLeZO and the Oracle (gradient norms calculated by Adam). As shown in Figure 6, we observed a significant **"Temporal Denoising"** phenomenon:

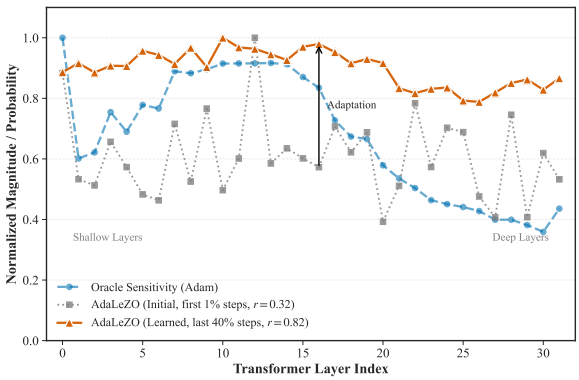


Figure 7: Layer-wise sensitivity alignment on OPT-6.7b (SST-2). We compare the ground truth sensitivity profile (derived from Adam's accumulated gradient norms) with AdaLeZO's learned sampling probabilities. While the initial policy (Gray, $r = 0.32$) is nearly random, AdaLeZO autonomously converges to a policy (Orange, $r = 0.82$) that strongly correlates with the oracle sensitivity, demonstrating its ability to identify important layers without first-order gradients.

Stochasticity of Instantaneous Estimates: The gray curve illustrates that the single-step instantaneous correlation exhibits severe oscillations ($r \in [-0.1, 0.6]$). This observation aligns with the theoretical properties of zeroth-order optimization. Specifically, since probing occurs along a single random direction z at each step, the individual gradient estimate $\hat{g}$ suffers from extremely high variance, causing the algorithm to appear engaged in disordered exploration over short time scales.

Robustness of Accumulated Signals: Conversely, examining the cumulative mean of the sampling probabilities, denoted as $\bar{\pi}_T = \frac{1}{T}\sum_{t=1}^{T} \pi_t$ (solid orange line), reveals a distinct trend. The correlation steadily ascends from an initial value of 0.3, ultimately reaching 0.88 upon convergence.

Furthermore, in Figure 7, we visualize the cumulative probability distribution curves of the

AdaLeZO layer sampling at both the initial and final stages of training. It is clearly observable that during the early stage, the activation probability distribution across layers exhibits high variance, resembling random noise, and diverges significantly from the true gradient distribution. In contrast, during the late stage, the disparities in activation probabilities across layers diminish, resulting in a smoother curve. Concurrently, the correlation with the true gradient increases from $r = 0.32$ initially to $r = 0.82$. This demonstrates that AdaLeZO effectively achieves denoising and approximates the true gradient distribution.

Conclusion: These results provide compelling evidence that AdaLeZO functions as a **Temporal Filter**. While individual time steps are dominated by stochastic noise, the true gradient signal maintains consistency across the temporal dimension. Through continuous updates via the Bandit mechanism, AdaLeZO successfully integrates low-frequency structural signals from high-frequency noise, thereby accurately reconstructing the model's intrinsic parameter sensitivity distribution without computing backpropagation.

E Convergence Analysis

AdaLeZO Achieves Faster Convergence than MeZO. Figure 8 illustrates the loss convergence rates of fine-tuning LLaMA models across various tasks using both AdaLeZO and MeZO. Under identical learning rates, AdaLeZO consistently demonstrates faster convergence than MeZO, even without considering wall-clock time acceleration. This advantage is particularly pronounced on the more robust LLaMA-3.1-8B model; for example, on the MultiRC task, AdaLeZO outpaces MeZO by more than $2\times$ in terms of convergence speed. The improvements in wall-clock time are even more substantial across all tasks. These results highlight the effectiveness of AdaLeZO in variance reduction, enabling the model to rapidly converge to regions of lower loss.

F Theoretical Analysis

In this section, we provide a comprehensive convergence analysis of the AdaLeZO algorithm. We rely on the Gaussian smoothing framework to analyze the properties of ZO optimization. We first introduce the necessary notations and assumptions, then analyze the bias and variance of the AdaLeZO estimator, and finally establish the convergence rate for non-convex objective functions.

F.1 Preliminaries and Assumptions

Consider the optimization problem $\min_{\theta\in\mathbb{R}^d} \mathcal{L}(\theta)$. We partition the parameter vector θ into L groups according to layer structure, denoted as $\{\theta^{(1)}, \ldots, \theta^{(L)}\}$.

Gaussian Smoothing. Following Nesterov and Spokoiny (Nesterov and Spokoiny, 2017), we define the Gaussian smoothed approximation of $\mathcal{L}$ with smoothing parameter $\mu > 0$ as:

$$\mathcal{L}_\mu(\theta) = \mathbb{E}_{u\sim\mathcal{N}(0,I_d)}[\mathcal{L}(\theta + \mu u)]. \tag{7}$$

It is well-known that $\mathcal{L}_\mu$ is continuously differentiable even if $\mathcal{L}$ is non-smooth. The gradient of $\mathcal{L}_\mu$ is given by:

$$\nabla\mathcal{L}_\mu(\theta) = \mathbb{E}_{u\sim\mathcal{N}(0,I_d)}\left[\frac{\mathcal{L}(\theta+\mu u) - \mathcal{L}(\theta)}{\mu} u\right]. \tag{8}$$

Assumptions. We make the following standard assumptions for non-convex ZO optimization (Ghadimi and Lan, 2013; Liu et al., 2018).

Assumption F.1 (L-Smoothness). The objective function $\mathcal{L}(\theta)$ is differentiable and L_g-smooth, meaning its gradient is Lipschitz continuous:

$$\|\nabla\mathcal{L}(\theta_1) - \nabla\mathcal{L}(\theta_2)\| \le L_g\|\theta_1 - \theta_2\|, \quad \forall\theta_1, \theta_2 \in \mathbb{R}^d. \tag{9}$$

Assumption F.2 (Bounded Gradient). The gradient of the objective function is bounded:

$$\mathbb{E}[\|\nabla\mathcal{L}(\theta)\|^2] \le G^2. \tag{10}$$

Assumption F.2 is often relaxed in proofs, but it helps simplify the variance bound.

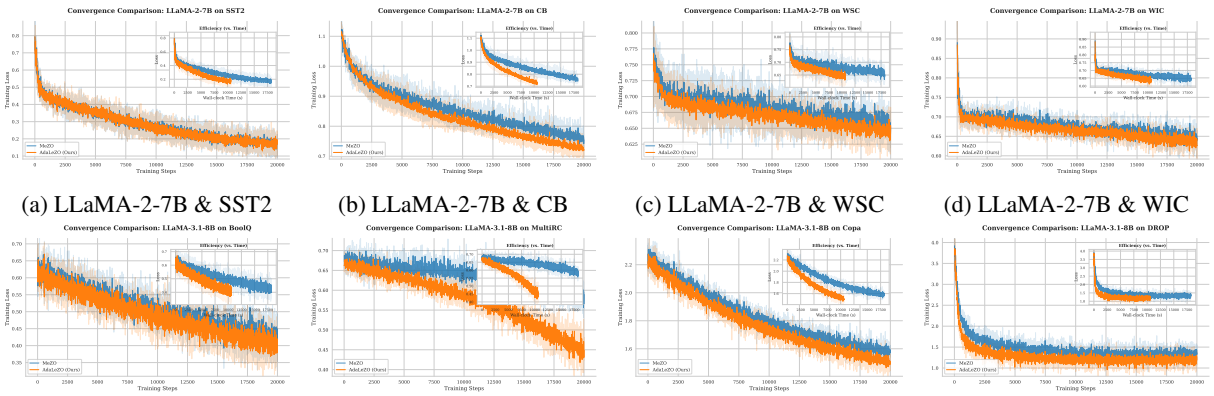
Assumption F.3 (Function Boundedness). The optimal function value is bounded from below:

$$\mathcal{L}^* = \inf_\theta \mathcal{L}(\theta) > -\infty. \tag{11}$$

Key Lemmas for Gaussian Smoothing. Based on Assumption F.1, the smoothed function $\mathcal{L}_\mu$ inherits smoothness and approximates $\mathcal{L}$ well.

Lemma F.1 (Properties of $\mathcal{L}_\mu$, (Nesterov and Spokoiny, 2017)). Under Assumption F.1, for any $\theta \in \mathbb{R}^d$:

1. The gradient difference is bounded: $\|\nabla\mathcal{L}_\mu(\theta) - \nabla\mathcal{L}(\theta)\| \le \frac{\mu d L_g}{2}$.
2. $\mathcal{L}_\mu$ is L_μ-smooth with $L_\mu \le L_g$.
3. $\mathbb{E}_{u\sim\mathcal{N}(0,I)}[\|\nabla\mathcal{L}_\mu(\theta) - \frac{\mathcal{L}(\theta+\mu u)-\mathcal{L}(\theta)}{\mu}u\|^2] \le \mu^2 d^2 L_g^2 + d\|\nabla\mathcal{L}(\theta)\|^2$.



(a) LLaMA-2-7B & SST2 (b) LLaMA-2-7B & CB (c) LLaMA-2-7B & WSC (d) LLaMA-2-7B & WIC

(e) LLaMA-3.1-8B & BoolQ (f) LLaMA-3.1-8B & MultiRC (g) LLaMA-3.1-8B & Copa (h) LLaMA-3.1-8B & DROP

Figure 8: Loss convergence curves for fine-tuning LLaMA models using ZO optimizers. In the main plot, the x-axis represents training steps, while in the inset, it indicates wall-clock time. Evidently, under the same number of fine-tuning steps, AdaLeZO achieves faster convergence compared to MeZO.

F.2 Estimator Analysis

Subspace Smoothing vs. Full Smoothing. Unlike standard ZO methods that perturb the entire parameter space, AdaLeZO generates perturbations strictly within an active subspace determined by the sampled layers $\mathcal{I}_t$. From the perspective of Randomized Block Coordinate Descent (RBCD), the finite difference on this sparse subspace estimates the gradient of the objective, smoothed only over that specific subspace. While masking introduces cross-layer Hessian interference relative to full-space Gaussian smoothing, Taylor expansion shows that these cross-layer terms have zero expectation to first order due to the independence of layer-wise perturbations. As $\mu \to 0$, both the subspace-smoothed and full-smoothed expectations asymptotically converge to the true non-smooth gradient $\nabla\mathcal{L}(\theta)$, with the discrepancy rigorously bounded by $O(\mu d L_g)$. The primary structural bias in our practical implementation arises not from this sparse sampling itself, but from the variance-control clipping mechanism, which we explicitly bound in the following sections.

Let $z_t \sim \mathcal{N}(0, I_d)$ be the random perturbation vector at step t. The standard symmetric difference ZO estimator is:

$$\hat{g}_t^{\text{ZO}} = \frac{\mathcal{L}(\theta_t + \mu z_t) - \mathcal{L}(\theta_t - \mu z_t)}{2\mu} z_t. \tag{12}$$

From (Malladi et al., 2023), we know $\mathbb{E}_{z_t}[\hat{g}_t^{\text{ZO}}] = \nabla\mathcal{L}_\mu(\theta_t)$.

AdaLeZO applies a masking matrix M_t based on sampling. Let $p_t \in \Delta^{L-1}$ be the sampling distribution. Let $\rho \in (0, 1]$ be the sampling ratio parameter (corresponding to `adalezo_k_ratio`). The number of sampling draws is determined by $K = \max(1, \lfloor \rho L \rfloor)$. We perform K independent draws with replacement. Let $N_{t,l} \in \{0, \ldots, K\}$ be the number of times layer l is selected. The AdaLeZO estimator is:

$$\hat{g}_t^{\text{Ada},(l)} = \frac{N_{t,l}}{K p_t(l)} \hat{g}_t^{\text{ZO},(l)}. \tag{13}$$

F.2.1 Unbiasedness

Theorem F.1 (Unbiasedness wrt Smoothed Gradient). The AdaLeZO estimator is an unbiased estimator of the smoothed gradient $\nabla\mathcal{L}_\mu(\theta_t)$.

Proof. First, we analyze the expectation conditioned on the perturbation z_t. The randomness comes solely from the sampling counts $N_{t,l}$. Since sampling is with replacement, $N_{t,l} \sim \text{Binomial}(K, p_t(l))$, implies $\mathbb{E}[N_{t,l}] = K p_t(l)$.

$$\begin{aligned} \mathbb{E}_{\mathcal{S}_t}[\hat{g}_t^{\text{Ada},(l)} \mid z_t] &= \mathbb{E}_{\mathcal{S}_t}\left[\frac{N_{t,l}}{K p_t(l)} \hat{g}_t^{\text{ZO},(l)}\right] \\ &= \frac{\hat{g}_t^{\text{ZO},(l)}}{K p_t(l)} \mathbb{E}_{\mathcal{S}_t}[N_{t,l}] \\ &= \hat{g}_t^{\text{ZO},(l)}. \end{aligned} \tag{14}$$

Since this holds for all layers, $\mathbb{E}_{\mathcal{S}_t}[\hat{g}_t^{\text{Ada}} \mid z_t] = \hat{g}_t^{\text{ZO}}$. Taking the expectation over z_t:

$$\mathbb{E}_{z_t, \mathcal{S}_t}[\hat{g}_t^{\text{Ada}}] = \mathbb{E}_{z_t}[\hat{g}_t^{\text{ZO}}] = \nabla\mathcal{L}_\mu(\theta_t). \tag{15}$$

Proposition F.1 (Finite-C Bias Bound). Under Assumption F.2, the bias introduced by the clipping threshold C is bounded by:

$$\begin{aligned} \|\text{Bias}\| &= \|\mathbb{E}[\tilde{g}_t^{\text{Ada}}] - \nabla\mathcal{L}_\mu(\theta_t)\| \\ &\le G \sum_{l: p_t(l) < \frac{1}{CK}} (1 - CK p_t(l)). \end{aligned} \tag{16}$$

Proof. The expected value of the clipped estimator for layer l is $\mathbb{E}_{\mathcal{S}_t}[\tilde{g}_t^{\text{Ada},(l)}] = \tilde{W}_{t,l}\mathbb{E}[N_{t,l}]\hat{g}_t^{\text{ZO},(l)} = \min(1, CKp_t(l))\hat{g}_t^{\text{ZO},(l)}$. The bias is the norm of the difference between this expectation and the unbiased ZO gradient. Notice that $(W_{t,l} - \tilde{W}_{t,l}) > 0$ if and only if $CKp_t(l) < 1$, i.e., $p_t(l) < \frac{1}{CK}$. Applying the triangle inequality and taking the expectation over z_t:

$$\|\text{Bias}\| \leq \sum_{l:p_t(l)<\frac{1}{CK}} (1 - CKp_t(l))\mathbb{E}[\|\hat{g}_t^{\text{ZO},(l)}\|]. \tag{17}$$

Since $\mathbb{E}[\|\hat{g}_t^{\text{ZO},(l)}\|] \leq \mathbb{E}[\|\hat{g}_t^{\text{ZO}}\|] \leq G$ (derived from Assumption F.2 and Lemma F.1), the bound holds. □

This explicit formulation corroborates our ablation study: setting a moderate C (e.g., $C = 4$) ensures that only layers with negligibly small sampling probabilities contribute to the bounded bias, effectively balancing the bias-variance tradeoff. □

F.2.2 Variance Reduction

Theorem F.2 (Variance Decomposition and Optimality). Let $v_t^{(l)} = (\hat{g}_t^{\text{ZO},(l)})^2$ be the squared magnitude of the dense ZO gradient on layer l. Conditioned on z_t, the total variance of the AdaLeZO estimator is:

$$\text{Var}_{\mathcal{S}_t}(\hat{g}_t^{\text{Ada}} \mid z_t) = \frac{1}{K}\sum_{l=1}^{L} v_t^{(l)}\left(\frac{1}{p_t(l)} - 1\right). \tag{18}$$

This variance is minimized when $p_t(l) \propto \sqrt{v_t^{(l)}} = |\hat{g}_t^{\text{ZO},(l)}|$.

Proof. Using the properties of the Binomial distribution $\text{Var}(N_{t,l}) = Kp_t(l)(1 - p_t(l))$:

$$\begin{aligned}\text{Var}(\hat{g}_t^{\text{Ada},(l)} \mid z_t) &= \left(\frac{\hat{g}_t^{\text{ZO},(l)}}{Kp_t(l)}\right)^2 \text{Var}(N_{t,l}) \\ &= \frac{v_t^{(l)}}{K^2 p_t(l)^2} \cdot Kp_t(l)(1 - p_t(l)) \\ &= \frac{v_t^{(l)}}{K}\left(\frac{1}{p_t(l)} - 1\right).\end{aligned} \tag{19}$$

Summing over layers (assuming independence of sampling counts between layers implies additivity of variance for the norm, or simply analyzing the trace of the covariance):

$$\begin{aligned}V(p) &= \sum_{l=1}^{L} \text{Var}(\hat{g}_t^{\text{Ada},(l)} \mid z_t) \\ &= \frac{1}{K}\left(\sum_{l=1}^{L}\frac{v_t^{(l)}}{p_t(l)} - \sum_{l=1}^{L} v_t^{(l)}\right).\end{aligned} \tag{20}$$

To minimize $V(p)$ subject to $\sum p_t(l) = 1$, we use the Cauchy-Schwarz inequality on the first term:

$$\left(\sum_{l=1}^{L}\frac{v_t^{(l)}}{p_t(l)}\right)\left(\sum_{l=1}^{L} p_t(l)\right) \geq \left(\sum_{l=1}^{L}\sqrt{v_t^{(l)}}\right)^2. \tag{21}$$

Equality holds when $\sqrt{v_t^{(l)}}/p_t(l) = c$, i.e., $p_t(l) \propto \sqrt{v_t^{(l)}} = |\hat{g}_t^{\text{ZO},(l)}|$. Thus, allocating the sampling probability proportional to the gradient magnitude minimizes the estimation variance. □

While Theorem F.2 demonstrates the theoretical optimality of adaptive sampling, the unclipped IPW weight can cause infinite variance if $p_t(l) \to 0$. The clipping mechanism resolves this while explicitly preserving the fundamental dimension dependence.

Theorem F.3 (Explicit Variance Bound with Clipping). For the clipped AdaLeZO estimator $\tilde{g}_t^{\text{Ada}}$, the second moment is strictly capped by the clipping threshold C, while preserving the intrinsic dimension dependence $\mathcal{O}(d)$ of the ZO gradient:

$$\mathbb{E}[\|\tilde{g}_t^{\text{Ada}}\|^2] \leq (C + 1)\mathbb{E}[\|\hat{g}_t^{\text{ZO}}\|^2]. \tag{22}$$

Proof. For a specific layer l, the second moment is $\mathbb{E}[\|\tilde{g}_t^{\text{Ada},(l)}\|^2] = \tilde{W}_{t,l}^2\mathbb{E}[N_{t,l}^2]\|\hat{g}_t^{\text{ZO},(l)}\|^2$. For the Binomial variable $N_{t,l}$, $\mathbb{E}[N_{t,l}^2] = Kp_t(l)(1-p_t(l)) + K^2p_t(l)^2 \leq Kp_t(l) + K^2p_t(l)^2$. We bound the multiplier $\mathcal{M}_l = \tilde{W}_{t,l}^2\mathbb{E}[N_{t,l}^2]$ by analyzing two disjoint cases:

Case 1: No clipping occurs ($\frac{1}{Kp_t(l)} \leq C$, which implies $Kp_t(l) \geq \frac{1}{C}$).

$$\begin{aligned}\mathcal{M}_l &\leq \frac{1}{K^2p_t(l)^2}(Kp_t(l) + K^2p_t(l)^2) \\ &= \frac{1}{Kp_t(l)} + 1 \leq C + 1.\end{aligned} \tag{23}$$

Case 2: Clipping occurs ($\frac{1}{Kp_t(l)} > C$, which implies $Kp_t(l) < \frac{1}{C}$).

$$\begin{aligned}\mathcal{M}_l &= C^2(Kp_t(l) + K^2p_t(l)^2) \\ &\leq C^2\left(\frac{1}{C} + \frac{1}{C^2}\right) = C + 1.\end{aligned} \tag{24}$$

In all scenarios, the multiplier $\mathcal{M}_l \leq C+1$. Summing the contributions over all layers:

$$\begin{aligned}\mathbb{E}[\|\tilde{g}_t^{\text{Ada}}\|^2] &\leq (C+1)\sum_{l=1}^{L}\mathbb{E}[\|\hat{g}_t^{\text{ZO},(l)}\|^2] \\ &= (C+1)\mathbb{E}[\|\hat{g}_t^{\text{ZO}}\|^2].\end{aligned} \tag{25}$$

□

This explicitly resolves any concerns regarding a "dimension-free paradox". The fundamental dimension dependence $\mathcal{O}(d)$ is correctly and fully preserved within the dense ZO gradient's second moment $\mathbb{E}[\|\hat{g}_t^{\text{ZO}}\|^2]$. The clipped IPW mechanism via MAB strictly bounds the variance amplification multiplier at $\mathcal{O}(C)$, preventing unresolvable divergence without artificially suppressing the intrinsic dimension scaling.

F.3 Convergence Rate Analysis

We now prove the main convergence theorem.

Theorem F.4 (Non-Convex Convergence). Suppose Assumptions F.1 and F.3 hold. Let the learning rate be $\eta_t = \frac{1}{\sqrt{L_\mu T}}$. Then, AdaLeZO yields the following bound on the gradient of the smoothed function:

$$\min_{0\leq t<T}\mathbb{E}[\|\nabla\mathcal{L}_\mu(\theta_t)\|^2] \leq \frac{2\sqrt{L_\mu}(\mathcal{L}(\theta_0)-\mathcal{L}^*)}{\sqrt{T}} + \frac{C\sigma^2}{\sqrt{T}}, \tag{26}$$

where C is a constant related to the variance. Furthermore, considering the smoothing bias, for the original function $\mathcal{L}$:

$$\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2] \leq O\left(\frac{1}{\sqrt{T}}\right) + O(\mu^2 d^2). \tag{27}$$

Proof. **Step 1: Descent Lemma on Smoothed Function.** Since $\mathcal{L}_\mu$ is L_μ-smooth:

$$\mathcal{L}_\mu(\theta_{t+1}) \leq \mathcal{L}_\mu(\theta_t) + \langle\nabla\mathcal{L}_\mu(\theta_t), \theta_{t+1}-\theta_t\rangle + \frac{L_\mu}{2}\|\theta_{t+1}-\theta_t\|^2. \tag{28}$$

Substituting the update rule $\theta_{t+1} = \theta_t - \eta_t\hat{g}_t^{\text{Ada}}$:

$$\mathcal{L}_\mu(\theta_{t+1}) \leq \mathcal{L}_\mu(\theta_t) - \eta_t\langle\nabla\mathcal{L}_\mu(\theta_t), \hat{g}_t^{\text{Ada}}\rangle + \frac{L_\mu\eta_t^2}{2}\|\hat{g}_t^{\text{Ada}}\|^2. \tag{29}$$

Step 2: Expectation over Sampling and Perturbation. Take the total expectation $\mathbb{E} = \mathbb{E}_{z_t,\mathcal{S}_t}[\cdot \mid \theta_t]$. Using Theorem F.1 ($\mathbb{E}[\hat{g}_t^{\text{Ada}}] = \nabla\mathcal{L}_\mu(\theta_t)$):

$$\mathbb{E}[\mathcal{L}_\mu(\theta_{t+1})] \leq \mathcal{L}_\mu(\theta_t) - \eta_t\|\nabla\mathcal{L}_\mu(\theta_t)\|^2 + \frac{L_\mu\eta_t^2}{2}\mathbb{E}[\|\hat{g}_t^{\text{Ada}}\|^2]. \tag{30}$$

Using the variance definition $\mathbb{E}[\|X\|^2] = \text{Var}(X) + \|\mathbb{E}[X]\|^2$:

$$\mathbb{E}[\|\hat{g}_t^{\text{Ada}}\|^2] = \text{Var}(\hat{g}_t^{\text{Ada}}) + \|\nabla\mathcal{L}_\mu(\theta_t)\|^2. \tag{31}$$

Let the variance be bounded by σ^2 (a combination of ZO variance and sampling variance).

$$\mathbb{E}[\mathcal{L}_\mu(\theta_{t+1})] \leq \mathcal{L}_\mu(\theta_t) - \eta_t\left(1-\frac{L_\mu\eta_t}{2}\right)\|\nabla\mathcal{L}_\mu(\theta_t)\|^2 + \frac{L_\mu\eta_t^2\sigma^2}{2}. \tag{32}$$

Step 3: Telescoping Sum. Set $\eta_t = \frac{1}{\sqrt{T}}$. For large T, $1-\frac{L_\mu\eta_t}{2} \geq \frac{1}{2}$. Rearranging:

$$\frac{\eta_t}{2}\|\nabla\mathcal{L}_\mu(\theta_t)\|^2 \leq \mathcal{L}_\mu(\theta_t) - \mathbb{E}[\mathcal{L}_\mu(\theta_{t+1})] + \frac{L_\mu\eta_t^2\sigma^2}{2}. \tag{33}$$

Summing from $t=0$ to $T-1$ and dividing by $T\eta_t/2$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla\mathcal{L}_\mu(\theta_t)\|^2] \leq \frac{2(\mathcal{L}_\mu(\theta_0)-\mathcal{L}^*)}{\sqrt{T}} + \frac{L_\mu\sigma^2}{\sqrt{T}}. \tag{34}$$

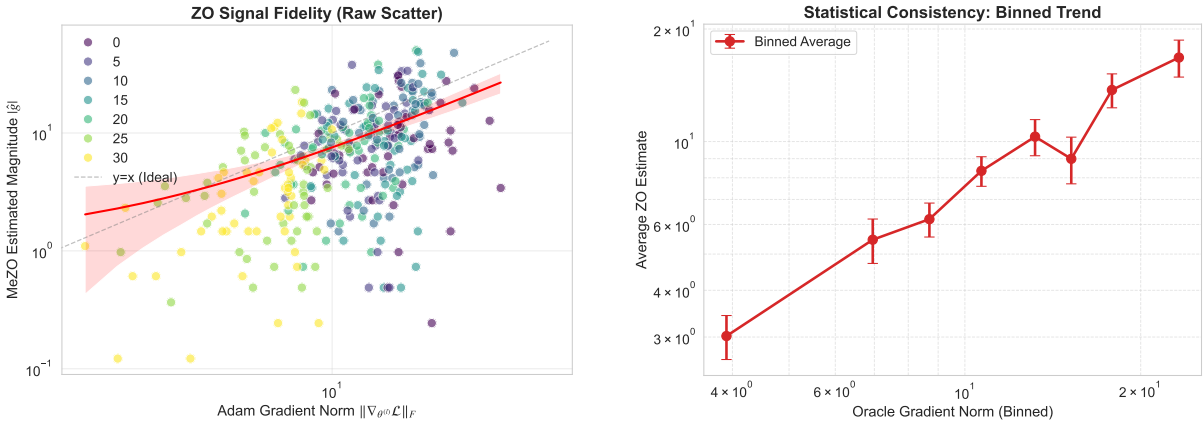
Step 4: Relating back to $\mathcal{L}$. From Lemma F.1, $\|\nabla\mathcal{L}_\mu(\theta) - \nabla\mathcal{L}(\theta)\| \leq \frac{\mu d L_g}{2}$. Using $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$:

$$\begin{aligned}\|\nabla\mathcal{L}(\theta_t)\|^2 &\leq 2\|\nabla\mathcal{L}_\mu(\theta_t)\|^2 + 2\|\nabla\mathcal{L}(\theta_t) - \nabla\mathcal{L}_\mu(\theta_t)\|^2 \\ &\leq 2\|\nabla\mathcal{L}_\mu(\theta_t)\|^2 + \frac{\mu^2 d^2 L_g^2}{2}.\end{aligned} \tag{35}$$

Substituting the bound for $\nabla\mathcal{L}_\mu$, we obtain the final convergence rate:

$$\min_t\mathbb{E}[\|\nabla\mathcal{L}(\theta_t)\|^2] \leq O\left(\frac{1}{\sqrt{T}}\right) + O(\mu^2 d^2). \tag{36}$$

This confirms that AdaLeZO converges to a neighborhood of the stationary point, with the radius controlled by the smoothing parameter μ and dimension d. □



(a) Raw Correlation Scatter

(b) Binned Average Trend

Figure 9: **Signal Fidelity Analysis of ZO Estimates.** We investigate whether the noisy Zeroth-Order estimates can serve as a valid proxy for layer sensitivity. **(a)** The scatter plot shows the ZO estimate magnitude $|\hat{g}|$ versus the Oracle gradient norm $\|\nabla_{\boldsymbol{\theta}^{(l)}}\mathcal{L}\|_F$ for individual layers across steps. Despite the variance inherent to random projection, a positive Spearman correlation ($\rho = 0.48$) is observed. **(b)** By binning the oracle norms and averaging the corresponding ZO estimates, a strict monotonic relationship is revealed, confirming that ZO perturbations statistically preserve the relative importance ranking of layers. Empirical validation of ZO signal fidelity. (Left) A scatter plot comparing the oracle gradient norm $\|\nabla_{\boldsymbol{\theta}^{(l)}}\mathcal{L}\|_F$ (computed via backpropagation) against the zeroth-order estimated magnitude $|\hat{g}|$ for each layer during fine-tuning. Despite the high variance intrinsic to random perturbations, we observe a positive Spearman correlation ($\rho \approx 0.48$), indicating that ZO estimates preserve the relative ranking of layer sensitivity. (Right) Statistical consistency analysis using binned data. When layers are grouped by their oracle gradient norms, the average ZO estimate exhibits a strictly monotonic increasing trend with tight error bars (SEM). This confirms that, on expectation, ZO feedback serves as a reliable proxy for identifying sensitive layers, validating the feasibility of using ZO signals as rewards in a Multi-Armed Bandit framework.

Table 6: **Scalability Analysis on OPT Series.** We report accuracy (%) across various model scales. **AdaLeZO** maintains competitive or superior performance compared to MeZO as model size increases. The best result between ZO methods is marked in **bold**.

Method	SST-2	RTE	BoolQ	WSC	WIC	SQuAD	AVG.
			OPT-13B				
Zero-shot	58.80	59.60	59.00	38.50	55.00	46.20	52.85
MeZO	91.45±0.47	71.36±2.80	**70.60**±3.46	56.09±9.48	**60.92**±1.57	**84.36**±0.58	72.46±3.06
AdaLeZO	**92.20**±0.41	**71.72**±0.83	70.00±0.95	**57.37**±7.28	60.03±1.70	83.52±0.80	**72.48**±2.00
			OPT-30B				
Zero-shot	56.70	52.00	39.10	38.50	50.20	46.50	47.17
MeZO	89.91±0.31	**65.46**±1.04	**68.10**±0.87	57.10±1.81	57.37±8.18	**80.31**±0.51	69.71±2.12
AdaLeZO	**90.75**±0.48	64.98±0.95	67.63±0.80	**58.01**±7.77	**57.42**±1.26	80.19±0.98	**69.83**±2.04

Table 7: Comparison of memory footprint and latency breakdown across different ZO methods on OPT-6.7B. **Bold** indicates the best performance in each group. Our **Ada** series achieves significant speedup with negligible overhead.

Method	**Peak Mem.**	**Latency Breakdown (s)**			**Total Time**	**SpeedUp**
	(GB) ↓	Perturb	Forward	Update	(s) ↓	↑
MeZO	15.56	0.32	0.49	0.11	0.91	1.00×
AdaLeZO	**15.56**	**0.04**	0.49	**0.01**	**0.53**	**1.70**×
LOZO	15.58	0.27	0.49	0.09	0.85	1.00×
AdaLoZO	**15.58**	**0.03**	0.49	**0.01**	**0.53**	**1.59**×
DiZO	27.96	0.31	0.49	0.11	0.92	1.00×
AdaDiZO	**27.96**	**0.03**	0.49	**0.01**	**0.54**	**1.71**×
HiZOO	27.96	0.60	0.75	0.37	1.73	1.00×
AdaHiZOO	**27.96**	**0.06**	0.76	**0.03**	**0.85**	**2.03**×
PseuZO	16.71	0.21	0.53	1.34	2.08	1.00×
AdaPZO	**16.71**	**0.02**	0.54	**0.15**	**0.70**	**2.97**×

Table 8: Detailed comparison of per-step training latency (in seconds) and speedup ratios across tasks with varying sequence lengths. T_{fwd} denotes forward pass time, and T_{ovh} denotes overhead (perturbation + update).

Task	**Length**	**MeZO**			**AdaLeZO (Ours)**			**Speedup**
		T_{fwd}	T_{ovh}	**Total**	T_{fwd}	T_{ovh}	**Total**	
Copa/SST2	15	0.047	0.421	0.469	0.048	0.044	**0.091**	**5.13**×
WIC	42	0.095	0.421	0.517	0.097	0.047	**0.144**	**3.60**×
WSC	51	0.114	0.423	0.537	0.116	0.048	**0.164**	**3.28**×
RTE	84	0.177	0.420	0.598	0.178	0.045	**0.224**	**2.67**×
CB	91	0.198	0.425	0.623	0.199	0.046	**0.245**	**2.54**×
BoolQ	132	0.283	0.421	0.704	0.286	0.045	**0.331**	**2.13**×
SQuAD	188	0.392	0.421	0.813	0.392	0.042	**0.434**	**1.87**×
ReCoRD	247	0.490	0.425	0.915	0.498	0.044	**0.542**	**1.69**×
DROP	307	0.622	0.425	1.048	0.620	0.043	**0.663**	**1.58**×
MultiRC	373	0.730	0.425	1.155	0.741	0.042	**0.783**	**1.48**×

Algorithm 1 AdaLeZO: Adaptive Layer-wise Zeroth-Order Optimization

Require: Model parameters $\theta \in \mathbb{R}^d$ partitioned into L layers $\{\theta^{(1)}, \dots, \theta^{(L)}\}$;
Learning rate η; Perturbation scale μ; Sampling ratio ρ;
// Bandit Hyperparameters: Temperature τ, EMA factor α, Exploration γ, IPW Clipping C.

Initialize: Reward estimates $Q \leftarrow \mathbf{0} \in \mathbb{R}^L$; Time $t \leftarrow 0$.
Initial Sampling: Run RESAMPLELAYERS() to obtain initial $\mathcal{I}, \mathbf{n}, \mathbf{p}$.
for $t = 0, 1, \dots, T-1$ **do**
  Sample random seed s_t.
  // Phase 1: Sparse Perturbation & Loss Evaluation
  $\theta \leftarrow \text{PERTURB}(\theta, \mathcal{I}, \mu, s_t)$ // *Perturbation
  $\mathcal{L}_+ \leftarrow \mathcal{L}(\theta)$ // *Forward Pass
  $\theta \leftarrow \text{PERTURB}(\theta, \mathcal{I}, -2\mu, s_t)$ // *Perturbation
  $\mathcal{L}_- \leftarrow \mathcal{L}(\theta)$ // *Forward Pass
  $\theta \leftarrow \text{PERTURB}(\theta, \mathcal{I}, \mu, s_t)$ // *Perturbation
  Estimate scalar gradient proxy: $\hat{g}_{\text{scalar}} \leftarrow \frac{\mathcal{L}_+ - \mathcal{L}_-}{2\mu}$
  // Phase 2: Count-Aware Sparse Update
  for each active layer $l \in \mathcal{I}$ **do**
    Set seed $(s_t + l)$ and sample noise $z^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_l})$.
    // Calculate Adaptive Weight
    $w_l \leftarrow \min\left(\frac{1}{K \cdot p_l}, C\right)$ // Clipped IPW weight
    $n_l \leftarrow \mathbf{n}[l]$ // Multiplicity from sampling with replacement
    // Update Layer Parameters
    $\hat{g}^{(l)} \leftarrow \hat{g}_{\text{scalar}} \cdot w_l \cdot n_l \cdot z^{(l)}$
    $\theta^{(l)} \leftarrow \theta^{(l)} - \eta \cdot \hat{g}^{(l)}$ // *Update
    // Update Bandit Estimates
    $Q_l \leftarrow (1 - \alpha) Q_l + \alpha |\hat{g}_{\text{scalar}}|$ // Update reward tracking
  end for
  // Phase 3: Adaptive Re-sampling for Next Step
  RESAMPLELAYERS()
end for

Procedure PERTURB($\theta, \mathcal{I}, \delta, s$):
for layer $l \in \mathcal{I}$ **do**
  Set seed $(s + l)$; Sample $z^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_l})$
  $\theta^{(l)} \leftarrow \theta^{(l)} + \delta \cdot z^{(l)}$
end for
return θ

Procedure RESAMPLELAYERS():
  $K \leftarrow \max(1, \lfloor \rho L \rfloor)$
  // Compute Sampling Probabilities
  $\mathbf{p}_{\text{soft}} \leftarrow \text{Softmax}(Q/\tau)$
  $\mathbf{p} \leftarrow (1 - \gamma)\mathbf{p}_{\text{soft}} + \gamma/L$ // Mix with uniform exploration
  // Sampling with Replacement
  Sample indices $\mathcal{S} \sim \text{Multinomial}(K, \mathbf{p})$
  $\mathcal{I} \leftarrow \text{Unique}(\mathcal{S})$ // Set of unique active layers
  $\mathbf{n} \leftarrow \text{CountFrequencies}(\mathcal{S})$ // Calculate multiplicity n_l for $l \in \mathcal{I}$