Improving Quantized Zeroth-Order Optimization through Reconstructed Low-Rank Structures

Fei Wang[1,2*], Shuai Xie[2*], Li Shen[3,4], Chao Xue[2], Ye Liu[1], Changxing Ding[1†]
[1]South China University of Technology [2]JD.com
[3]Shenzhen Campus of Sun Yat-sen University
[4]Center for AI Theoretical Foundation and Systems, Shenzhen Loop Area Institute
wang_fei2019@163.com, chxding@scut.edu.cn

Abstract

Fine-tuning large language models (LLMs) under stringent memory constraints remains a significant challenge. Recently, zeroth-order (ZO) optimizers have been employed to fine-tune quantized models, avoiding backpropagation and further reducing memory usage. However, quantized ZO optimizers often underperform compared to their full-precision counterparts, especially at low bit-widths. Motivated by concentrated gradient spectra in LLM fine-tuning, we investigate whether structured low-rank perturbations can improve quantized scale optimization. To address this, we propose the **L**ow-rank **Q**uantized **Z**eroth-**O**rder optimizer (LQZO), which enhances quantized ZO fine-tuning by leveraging the low-rank structure inherent in gradients. Our method introduces three key innovations: (1) A low-rank parameterization of gradient perturbations using structured matrix factorization. (2) A Binary-Aggregated Gaussian perturbation strategy that employs a Rademacher distribution to constrain higher-order moments and tighten the variance bound of gradient estimates, without additional loss evaluations. (3) An Isotropy-Enhanced Estimation technique that reshapes quantization scale groups into more isotropic structures to diversify exploration directions. We provide a convergence analysis for the independently refreshed base estimator, including finite-difference bias. The practical lazy-update and clipping variants are evaluated empirically. Extensive experiments demonstrate that LQZO improves average task scores over QZO by 2.0%–3.3% across quantization precisions. While extreme quantization inevitably leads to information loss, LQZO significantly improves the trainability of INT2 models and narrows the performance gap compared to prior quantized ZO methods. Furthermore, we demonstrate the cross-modality versatility of LQZO by validating its superior performance on Vision Transformers.[3]

1 Introduction

Large language models (LLMs) [1–4] have demonstrated exceptional performance across various natural language processing tasks [5–8]. However, their substantial parameter count renders full-parameter fine-tuning computationally intensive and memory-intensive, posing significant deployment challenges, especially on edge devices or under tight memory budgets [9–11]. Model quantization, which converts high-precision weights into low-bit representations, has emerged as a pivotal technique to alleviate memory pressure [12–15]. Nonetheless, efficiently fine-tuning quantized models for downstream tasks without incurring significant memory overhead remains a persistent challenge.

*Equal contribution. This work was done while Fei Wang was at JD Explore Academy.
†Corresponding author.
[3]Code is available at https://github.com/WangFei-2019/UnifiedZO.

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

Direct first-order updates to discrete quantized weights [16, 17] are complicated by non-differentiable rounding. First-order updates to continuous quantization scales remain possible when integer weights are fixed, but require backpropagation and its associated memory costs. Although Quantization-Aware Training [18] mitigates this issue by simulating quantization, it still requires storing full-precision weights and optimizer states in memory, thereby reducing memory efficiency. Recently, ZO optimization [19–22] has gained traction as a promising alternative. ZO methods estimate gradients solely via forward passes [23], circumventing backpropagation and its associated memory costs. However, directly applying ZO methods to quantized models is challenging. Perturbing the discrete quantized weights is often ineffective due to the rounding operation, while perturbing the dequantized weights disrupts the calibrated quantization grid. To address this, approaches such as QZO [24] propose directly perturbing the quantization scales, opening a new avenue for ZO fine-tuning of quantized LLMs [25, 26]. Despite this progress, existing quantized ZO methods suffer significant performance degradation as bit-width decreases, especially at INT2 [27].

Low-rank gradient compression [28], low-rank ZO methods [29, 30], and studies of restricted trainable subspaces [31, 32] motivate investigating structured perturbations for quantized scale optimization. The flatter QZO checkpoint-displacement spectrum in Fig. 3 motivates this investigation, but does not alone establish the cause of its performance gap.

To address this limitation, we propose the **L**ow-Rank **Q**uantized **Z**eroth-**O**rder Optimizer (LQZO). The core idea of LQZO is to explicitly reconstruct and exploit low-rank structures during gradient estimation, enabling more efficient and stable fine-tuning under low-bit quantization. Parameter-Efficient Fine-Tuning [33–35] introduces additional trainable parameters, whereas LQZO directly optimizes the existing quantization scales without introducing adapter modules into the deployed model. Our contributions are threefold: (1) **Low-rank parameterization of gradient perturbations**: We model perturbations via structured matrix factorization, i.e., as the product of two thin matrices. This reduces the number of random entries used to generate perturbations and restricts each update to a random low-rank matrix, while preserving the number of trainable scale parameters. (2) **Binary-Aggregated Gaussian Perturbation**: This sampling strategy draws one low-rank component from a Gaussian distribution and the other from a Rademacher distribution. It inherently constrains the higher-order moments of the perturbations, reducing the finite-difference moment bound and the exact single-block second moment without adding loss evaluations. (3) **Isotropy-Enhanced Estimation**: To combat the severe dimensional imbalance typical of quantization scale matrices, we introduce a geometric reshaping strategy. This balances the perturbation-space dimensions and improves empirical performance without increasing the parameter count.

We analyze asymptotic unbiasedness, second moments, and convergence of the independently refreshed base estimator with explicit finite-difference bias. Experiments with the practical lazy-update and clipping variants show that LQZO improves average scores over QZO across the tested LLM quantization levels, including INT2, and on quantized vision transformers (ViTs).

2 Related Work

Zeroth-order Optimization. ZO optimizers [19, 36–38] estimate gradients via forward passes without backpropagation, offering a memory-efficient alternative to first-order methods. MeZO [19] pioneers ZO fine-tuning for LLMs by integrating SPSA [23] with ZO-SGD [39]. Subsequent works improve efficiency through sparse perturbations [40], layer-wise updates [21], low-rank subspaces [29, 41], and second-order information [42]. However, these methods store weights in full precision. Combining ZO with quantization [25, 27, 26, 24] is a natural progression. Notably, QZO [24] directly perturbs quantization scales. Despite this, existing quantized ZO optimizers suffer significant performance degradation at low bit-widths and do not explicitly exploit the intrinsic low-rank structure of gradients.

Quantization and Low-Bit Adaptation. Quantization compresses LLMs by representing weights in low precision [43–47]. PTQ methods such as GPTQ [13], LLM.int8() [12], and SmoothQuant [14] map floating-point values to integers using full-precision scales, while codebook-based approaches [48, 49] extend compression to 2–3 bits. Fine-tuning quantized models is challenging because the quantization function is non-differentiable. QLoRA [34] bypasses this by appending full-precision low-rank adapters and relies on backpropagation through those adapters. Quantization-Aware Training [18] simulates quantization during training but uses backpropagation in full precision.

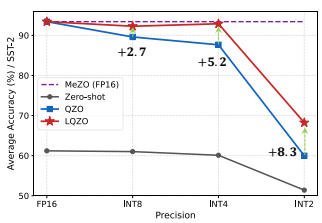


Figure 1: Performance on SST-2 at different bit widths. Arrows mark LQZO gains over QZO.

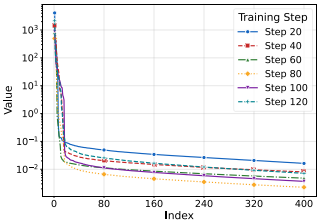


Figure 2: Weight-gradient spectrum for *fc2* in layer 31 of OPT-6.7B during Adam fine-tuning (top 400 of 4096 values).

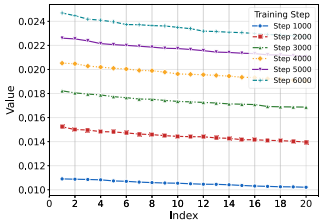


Figure 3: Top 20 of 512 singular values of QZO checkpoint displacements for *fc2* in layer 31, showing a flat spectrum in this setting.

ZO optimization offers a purely optimizer-level alternative. QZO [24] perturbs quantization scales without adding external parameters, yet ignores the low-rank gradient geometry.

Our Contribution. Unlike prior quantized ZO methods, LQZO reconstructs low-rank gradient structures via three innovations: low-rank parameterization of perturbations, Binary-Aggregated Gaussian perturbations that tighten the variance bound by constraining higher-order moments, and Isotropy-Enhanced Estimation that reshapes quantization groups to enhance directional diversity. These collectively enable efficient fine-tuning even at extreme quantization levels such as INT2.

3 Rethinking Quantized ZO Optimizers

The performance of fine-tuned LLMs on downstream tasks degrades significantly as quantization precision is reduced [27]. As illustrated by the blue curve in Fig. 1, at INT2 precision, the performance of models fine-tuned using QZO only marginally exceeds the zero-shot baseline. This demonstrates that the effectiveness of QZO in restoring model performance diminishes sharply at lower bit-widths, making it a critical challenge to enhance ZO optimizers for quantized models.

From an optimization perspective, the gradients of LLMs exhibit low-rank structures, a phenomenon that has been extensively studied and confirmed in numerous works [31, 32, 28]. Emphasizing low-rank relationships within LLMs during ZO fine-tuning can enhance the performance and convergence efficiency of the fine-tuned models [29]. For instance, Fig. 2 shows the low-rank structure of weight gradients during full-precision fine-tuning with Adam [17], a first-order optimizer. Fewer than 1% of the singular values are significant (i.e., exceed 0.1). With this established prior, we investigate a pivotal question:

Does the Quantized ZO optimizer effectively leverage this low-rank structure during fine-tuning?

To investigate this, we analyze checkpoint displacements during QZO fine-tuning. Fig. 3 presents the corresponding singular value distribution [4]. Unlike the highly skewed spectrum observed in full-precision fine-tuning (Fig. 2), the singular value spectrum of the QZO checkpoint displacements is remarkably flat and compressed, lacking any clear, dominant update directions. This motivates an investigation of whether structured perturbations can improve scale optimization. Since this comparison involves different parameterizations and compares instantaneous gradients with cumulative displacements, it cannot alone identify the cause of the spectral difference. Sec. H.1 provides complementary comparisons on the same scale parameters and on full-precision weights.

To address this limitation, we propose the Low-Rank Quantized ZO optimizer, which is specifically designed to enhance the performance of quantized ZO fine-tuning by explicitly exploiting the low-rank characteristics of gradient matrices.

[4] For the ZO spectra, we use differences between consecutive parameter checkpoints accumulated over multiple optimization steps. These displacements combine estimated updates and learning-rate effects along a changing trajectory. They are not instantaneous gradients at a fixed parameter value.

4 Methodology

In this section, we revisit SPSA [23] and ZO-SGD [19] and introduce the implementation of the Quantized ZO optimizer [24]. Then we introduce LQZO, a low-rank ZO algorithm specifically designed for quantized LLMs, and provide detailed insights into its technical implementation.

4.1 Preliminaries

4.1.1 Zeroth-order Optimization

ZO optimizers approximate gradients using finite-difference estimates, circumventing the need for backpropagation. MeZO combines a Gaussian two-point gradient estimator, inspired by the simultaneous-perturbation principle of SPSA [23], with a stochastic update rule [19].

SPSA [23] is an efficient stochastic optimization technique for multivariable systems, designed to solve high-dimensional optimization problems. It approximates gradients by perturbing all parameters simultaneously and using only two function evaluations. Considering an optimization step for LLMs, we define an objective function ℓ and a trainable parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$, where d denotes the dimension of the trainable parameters. The training dataset is defined as $\mathcal{D}$, from which a mini-batch of data $\mathcal{B}$ is drawn. The Gaussian two-point estimator used by MeZO is expressed as:

$$\widehat{\nabla}\ell(\boldsymbol{\theta};\mathcal{B}) := \frac{\ell(\boldsymbol{\theta}+\mu\boldsymbol{z};\mathcal{B}) - \ell(\boldsymbol{\theta}-\mu\boldsymbol{z};\mathcal{B})}{2\mu}\boldsymbol{z} \approx \boldsymbol{z}\boldsymbol{z}^\top\nabla\ell(\boldsymbol{\theta};\mathcal{B}),$$

where $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}_d)$, $\boldsymbol{I}_d$ is the $d \times d$ identity, and μ is a small positive perturbation scale.

Malladi et al. [19] extend the SGD method to the ZO optimization process by using the estimated gradient to update LLM parameters. Considering the t-th step in the training process, the ZO-SGD parameter update process can be expressed as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t\widehat{\nabla}\ell(\boldsymbol{\theta}_t;\mathcal{B}_t), \tag{1}$$

where $\mathcal{B}_t$ denotes the training data used at the t-th step and η_t is the learning rate. Through iterative estimation and updates, MeZO effectively fine-tunes LLMs on a dataset $\mathcal{D}$.

4.1.2 ZO Optimizer for Quantized LLMs

Quantization maps floating-point weights to the nearest data points using a scaling factor, significantly reducing the memory required for weight storage. During inference, dequantization is typically performed for matrix multiplication. For symmetric quantization, the quantization and dequantization of a weight w can be represented as:

$$\overline{w} = \lfloor\frac{w}{\delta}\rceil; \qquad \hat{w} = \delta \cdot \overline{w}, \tag{2}$$

where $\overline{w}$ is the quantized weight (e.g., an integer), $\hat{w}$ is the dequantized weight, and δ is the quantization scale shared by a group of weights. Let $\boldsymbol{w} \in \mathbb{R}^c$ denote a group of weights. Common scale choices include $\delta = \frac{\|\boldsymbol{w}\|_\infty}{2^{b-1}-1}$ for symmetric quantization and $\delta = \frac{\max(\boldsymbol{w})-\min(\boldsymbol{w})}{2^b-1}$ for asymmetric quantization, where b is the bit width. Asymmetric dequantization uses $\delta(q-z)$ with an integer code q and a fixed zero point z. Below, any fixed zero-point correction is absorbed into $\overline{w}$.

Applying MeZO directly to quantized weights is non-trivial. QZO [24] instead perturbs continuous scales while fixing the quantized coefficients $\overline{\boldsymbol{\theta}}$ and all non-quantized weights. We write $\mathcal{L}(\boldsymbol{\Delta};\mathcal{B}) := \ell(\mathcal{R}^{-1}(\boldsymbol{\Delta}) \odot \overline{\boldsymbol{\theta}};\mathcal{B})$ for the scale-space loss. Here $\mathcal{R}$ is a fixed reshape, the identity for QZO. The grouped scales are broadcast over their assigned weights before multiplication. Q-SPSA is

$$\widehat{\nabla}\mathcal{L}(\boldsymbol{\Delta};\mathcal{B}) := \frac{\mathcal{L}(\boldsymbol{\Delta}+\mu\boldsymbol{z};\mathcal{B}) - \mathcal{L}(\boldsymbol{\Delta}-\mu\boldsymbol{z};\mathcal{B})}{2\mu}\boldsymbol{z} \approx \boldsymbol{z}\boldsymbol{z}^\top\nabla\mathcal{L}(\boldsymbol{\Delta};\mathcal{B}). \tag{3}$$

Here $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}_{d'})$, with d' trainable scales. For a clipping threshold $C > 0$, clipping the scalar factor in the raw estimate $g_{\text{est}}\boldsymbol{z}$ gives

$$g_{\text{clip}} = \max(-C, \min(C, g_{\text{est}})), \tag{4}$$

and the clipped estimate is $\widehat{\nabla}^{\text{clip}}\mathcal{L} = g_{\text{clip}}\boldsymbol{z}$.

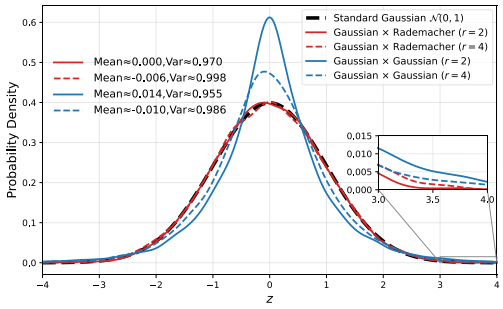


Figure 4: Empirical distribution of Z_{ij}/r comparing our Binary-Aggregated Gaussian Perturbation (Rademacher V) with Gaussian V, estimated from 10,000 independent samples. Here U, V, Z, r denote U_k, V_k, Z_k, r_k for one block.

4.2 Low-Rank Quantized Zeroth-Order Optimizer

4.2.1 Low-Rank Quantized SPSA

Let $\mathbf{\Delta}^{\text{orig}} = \{\Delta_k^{\text{orig}}\}_{k=1}^{\mathcal{K}}$ denote the stored scales, with block shape $M_k \times N_k$. We work with $\Delta_k = \mathcal{R}_k(\Delta_k^{\text{orig}}) \in \mathbb{R}^{m_k \times n_k}$, where $m_k = M_k/\gamma$ and $n_k = \gamma N_k$. $\gamma = 1$ gives no reshaping. Thus $\mathbf{\Delta} = \{\Delta_k\}_{k=1}^{\mathcal{K}}$ is always in perturbation coordinates and $\mathcal{K}$ is the number of blocks. We sample $U_k \in \mathbb{R}^{m_k \times r_k}$ and $V_k \in \mathbb{R}^{n_k \times r_k}$, with positive integer r_k, typically $r_k \ll \min\{m_k, n_k\}$. We define the perturbation in the quantization SPSA as a low-rank representation formed by the matrices U_k and V_k, denoted as $Z_k = U_k V_k^\top$. The rank of Z_k is at most r_k. Considering perturbations across all quantization scales of the quantized model and fixing all quantized weights, we define the Low-Rank Quantized SPSA (LR-Q-SPSA) in a macroscopic form:

$$\widehat{\nabla}\mathcal{L}(\mathbf{\Delta}; \mathcal{B}) := \frac{\mathcal{L}(\mathbf{\Delta} + \mu\mathbf{Z}; \mathcal{B}) - \mathcal{L}(\mathbf{\Delta} - \mu\mathbf{Z}; \mathcal{B})}{2\mu} \cdot \frac{\mathbf{Z}}{\boldsymbol{r}}, \tag{5}$$

where $\mathbf{Z} := \{Z_k\}_{k=1}^{\mathcal{K}}$, $\mathbf{U} := \{U_k\}_{k=1}^{\mathcal{K}}$, and $\mathbf{V} := \{V_k\}_{k=1}^{\mathcal{K}}$. Products are defined block by block. At a fixed parameter value, averaging over independently sampled $\mathbf{U}$ and $\mathbf{V}$, the factor $1/\boldsymbol{r}$ ($\boldsymbol{r} = \{r_k\}_{k=1}^{\mathcal{K}}$) makes the estimator asymptotically unbiased as $\mu \to 0$. A finite μ introduces an approximation bias, which is retained explicitly in the convergence analysis. The element-wise block scaling $\boldsymbol{r}$ means each Z_k is divided by its corresponding rank r_k. The low-rank constraint applies to each sampled perturbation, while all scale entries remain trainable. Refreshing factors changes the update subspaces, so accumulated updates need not retain a single perturbation's rank.

The factorization reduces the random entries per block from $m_k n_k$ to $(m_k + n_k)r_k$ for small rank, while preserving the number of trainable scales. Generating $\mathbf{U}$ and $\mathbf{V}$ incurs time overhead [21]. To mitigate this computational cost without compromising performance, we adopt an amortized subspace generation strategy named lazy updates [29]. We regenerate $\mathbf{V}$ every τ optimization steps and reuse it between updates, reducing random number generation costs.

The raw LR-Q-SPSA update applies step size $\eta_t^{\text{eff}} = 2^{4-b}\eta_t$, with base learning rate η_t:

$$\mathbf{\Delta}_{t+1} = \mathbf{\Delta}_t - \eta_t^{\text{eff}} \widehat{\nabla}\mathcal{L}(\mathbf{\Delta}_t; \mathcal{B}_t), \tag{6}$$

In App. G, we establish asymptotic unbiasedness and a second-moment bound for independently sampled LR-Q-SPSA, and derive a convergence guarantee that includes finite-difference bias. The guarantee concerns the unclipped base estimator with fresh factors at every step ($\tau = 1$). The practical algorithm in Alg. 1 additionally uses lazy updates and scalar clipping. These modifications are evaluated empirically and are not covered by that guarantee.

4.2.2 Binary-Aggregated Gaussian Perturbation

Standard ZO methods use Gaussian perturbations [19, 22, 21], while prior low-rank ZO methods sample both factors from Gaussian distributions [29]. For $[Z_k]_{ij} = \sum_{a=1}^{r_k} [U_k]_{ia}[V_k]_{ja}$, both constructions have zero mean and variance r_k, but their higher-order moments differ.

We sample U_k from a standard Gaussian distribution and V_k from a Rademacher distribution. The columns of U_k provide directions in $\mathbb{R}^{m_k}$, and rows of V_k combine them into columns of Z_k. Each marginal $[Z_k]_{ij}/r_k$ is exactly $\mathcal{N}(0, 1/r_k)$, as illustrated in Fig. 4. The joint matrix distribution still has dependence across entries.

At a fixed rank, this construction reduces the fourth and sixth marginal moments relative to Gaussian V_k. App. G gives the explicit moments and shows their effect on the finite-difference moment bound and exact single-block second moment. The sampling change requires no additional loss evaluations. Fig. 10 measures the overall reconstruction overhead.

4.2.3 Isotropy-Enhanced Estimation

Quantization methods like GPTQ often use large group sizes for compression, resulting in severely imbalanced stored scale matrices Δ_k^{orig} where $M_k \gg N_k$. This poses a fundamental challenge for low-rank perturbations in LQZO.

From a geometric perspective, the exploratory capability of a rank-r_k perturbation $Z_k = U_k V_k^\top \in \mathbb{R}^{m_k \times n_k}$ is inherently constrained by both its rank and the smaller dimension n_k: the column space of Z_k lies in a subspace of dimension at most $\min(r_k, n_k)$. Before reshaping, $m_k = M_k$ and $n_k = N_k$. When n_k is small, the effective directions explored by Z_k along the column space are severely limited, regardless of how large m_k is. For GPTQ group size 128, the quantization scales of OPT-6.7B have dimensions 4096×32, and a rank-2 perturbation occupies at most $2/32$ of the row-space dimension versus $2/4096$ of the column-space dimension, a $128\times$ coverage imbalance.

The single-block second moment in Eq. (23) contains the shape-dependent term $(\kappa_4^U - 1)n_k + (\kappa_4^V - 1)m_k$ at fixed product $m_k n_k$. For Gaussian U_k and V_k, both coefficients equal two, so balancing the dimensions minimizes this term. For the mixed Gaussian and Rademacher construction used by LQZO, the coefficients are $\kappa_4^U = 3$ and $\kappa_4^V = 1$, leaving $2n_k$. Thus minimizing $m_k + n_k$ is a geometric balancing criterion, rather than a minimizer of this mixed-distribution second moment. Its benefit for LQZO is supported empirically by the reshaping ablation.

To address this limitation, we propose *Isotropy-Enhanced Estimation*, which transforms imbalanced scale matrices into more balanced matrix shapes through reshaping. For each block, $\mathbb{E}[\mathrm{vec}(Z_k)\,\mathrm{vec}(Z_k)^\top] = r_k \boldsymbol{I}_{m_k n_k}$ already holds before reshaping. A fixed reshaping preserves this covariance, so its effect concerns matrix structure and higher-order dependence rather than correcting an anisotropic second moment. We choose γ to minimize the dimension sum:

$$\gamma_k^* \in \arg\min_{\gamma \in \mathcal{G}_k} \left(\frac{M_k}{\gamma} + \gamma N_k \right), \tag{7}$$

where $\mathcal{G}_k = \{2^j : j \in \mathbb{Z}_{\geq 0},\ 2^j \mid M_k\}$ enforces power-of-two reshaping and integer M_k/γ. The minimizer need not be unique. Our implementation uses a compatible shared $\gamma \in \bigcap_k \mathcal{G}_k$ selected by ablation. This per-block criterion motivates candidate values rather than requiring every block to attain its own minimum. This criterion balances the matrix aspect ratio while maintaining compatibility with the grouping layout. It does not imply a tighter second-moment bound for every choice of perturbation distributions. Reshaping is a linear isomorphism. We apply low-rank perturbations in the reshaped space and then map them back to the original matrix structure. This bijection preserves the number of trainable parameters and the dequantization map. It changes the structured perturbation distribution in the original coordinates and reorganizes the exploration subspaces.

4.3 Convergence Analysis

In LQZO, we focus on fine-tuning the quantization scales $\boldsymbol{\Delta}$, which transform fixed quantized weights $\overline{\boldsymbol{\theta}}$ to dequantized weights via $\mathcal{R}^{-1}(\boldsymbol{\Delta}) \odot \overline{\boldsymbol{\theta}}$. The optimization objective is:

$$\min_{\boldsymbol{\Delta}} f(\boldsymbol{\Delta}) := \mathbb{E}_{\mathcal{B}}[\mathcal{L}(\boldsymbol{\Delta}; \mathcal{B})]. \tag{8}$$

The integer weights $\overline{\boldsymbol{\theta}}$ remain fixed while the continuous scales $\boldsymbol{\Delta}$ are optimized. Thus dequantization is differentiable with respect to the scales. Scale updates do not involve rounding.

Reshaping preserves the Frobenius norm and inner product. For block collections, $\langle \mathbf{A}, \mathbf{B} \rangle := \sum_{k=1}^{\mathcal{K}} \mathrm{tr}(A_k^\top B_k)$ and $\|\mathbf{A}\|^2 := \langle \mathbf{A}, \mathbf{A} \rangle$. For matrices we use the Frobenius norm. We adopt the following assumptions for the independently refreshed base estimator:

Table 1: Entries with ± report the mean and standard deviation over three random seeds. Performance comparison of ZO optimizers for fine-tuning OPT-6.7B at various quantization levels. Numbers in parentheses represent the improvement of LQZO over QZO.

Precision	Task / Method	SST-2	RTE	CB	BoolQ	WiC	WSC	MultiRC	COPA	ReCoRD	SQuAD	DROP	Avg.
					classification				multiple choice		generation		
FP16	zero-shot	61.2	55.2	51.8	59.5	51.3	37.5	44.5	82.0	77.0	36.5	17.9	52.2
	MeZO	93.4±0.4	70.9±1.5	69.6±3.1	69.4±1.8	61.1±3.1	56.4±9.1	63.7±1.8	80.3±2.0	80.3±1.5	80.1±0.5	28.5±0.7	68.5±2.3
INT8	zero-shot	61.0	54.2	55.4	60.1	51.3	36.5	44.1	82.0	76.1	36.2	17.6	52.2
	QZO	89.6±1.2	60.9±1.8	68.5±1.0	66.0±1.1	58.0±2.4	55.5±10.6	58.6±1.7	81.3±3.1	79.2±2.0	78.4±1.0	27.9±0.6	65.8±2.3
	LQZO	92.3±1.2	64.4±3.5	72.6±2.1	67.1±1.0	60.0±1.5	60.9±3.9	62.9±2.0	80.3±0.6	78.9±1.5	79.1±1.4	27.6±1.0	67.8±1.8 (↑2.0)
INT4	zero-shot	60.1	53.8	51.8	59.1	51.4	36.5	45.1	79.0	76.3	35.9	18.0	51.5
	QZO	87.7±0.6	62.3±2.1	67.9±1.8	64.9±1.7	57.1±1.8	53.5±12.3	59.5±3.8	82.0±1.7	78.2±1.9	79.1±1.2	28.0±0.4	65.5±2.7
	LQZO	92.9±1.0	69.6±1.8	70.8±6.8	67.3±1.0	61.4±1.4	57.1±10.3	60.2±1.6	79.7±0.6	78.4±1.5	79.6±0.5	27.9±1.2	67.7±2.5 (↑2.2)
INT2	zero-shot	51.4	47.3	26.8	42.6	49.7	40.4	49.3	56.0	41.6	6.1	2.5	37.6
	QZO	60.0±0.9	53.4±1.9	55.4±0.0	57.6±3.6	50.9±2.0	50.3±10.6	54.1±2.7	67.3±2.9	57.8±1.4	44.3±2.4	9.6±1.6	51.0±2.7
	LQZO	68.2±1.0	53.2±1.5	68.5±4.5	58.5±2.5	50.9±1.4	53.2±4.9	54.0±3.1	68.7±3.1	58.7±1.3	46.5±1.3	16.8±0.7	54.3±2.3 (↑3.3)

Assumption 4.1 (L-Smoothness)**.** *For any $\mathcal{B}$, the function $\mathcal{L}(\mathbf{\Delta};\mathcal{B})$ is differentiable. Its gradient $\nabla\mathcal{L}(\mathbf{\Delta};\mathcal{B})$ is uniformly L-Lipschitz continuous:*

$$\|\nabla\mathcal{L}(\mathbf{\Delta};\mathcal{B}) - \nabla\mathcal{L}(\mathbf{\Delta}';\mathcal{B})\| \le L\|\mathbf{\Delta} - \mathbf{\Delta}'\|, \quad \forall \mathbf{\Delta}, \mathbf{\Delta}', \mathcal{B}, \tag{9}$$

where $\|\mathbf{\Delta}\| := \sqrt{\sum_{k=1}^{\mathcal{K}} \|\Delta_k\|^2}$ denotes the aggregated Frobenius norm.

Assumption 4.2 (Unbiased Gradient and Bounded Variance)**.** *For every fixed $\mathbf{\Delta}$, the stochastic gradient is unbiased and has bounded variance. At step t, the mini-batch $\mathcal{B}_t$ is sampled independently of the preceding optimization history:*

$$\mathbb{E}_{\mathcal{B}}[\nabla\mathcal{L}(\mathbf{\Delta};\mathcal{B})] = \nabla f(\mathbf{\Delta}), \mathbb{E}_{\mathcal{B}}\|\nabla\mathcal{L}(\mathbf{\Delta};\mathcal{B}) - \nabla f(\mathbf{\Delta})\|^2 \le \sigma^2. \tag{10}$$

Assumption 4.3 (Perturbation Matrix Properties)**.** *For the base-estimator analysis, the factors $\mathbf{U} := \{U_k\}_{k=1}^{\mathcal{K}}$ and $\mathbf{V} := \{V_k\}_{k=1}^{\mathcal{K}}$ are mutually independent across factors and blocks, independent of the mini-batch and optimization history, and independently refreshed at every step from fixed entry distributions. Entries within each factor are i.i.d., symmetric, zero-mean, and unit-variance, with finite sixth moments. We use a common rank $r_k = r \ge 1$ for every block in the second-moment and convergence bounds. In particular:*

$$\mathbb{E}[U_k^\top U_k] = m_k \boldsymbol{I}_{r_k}, \quad \mathbb{E}[U_k U_k^\top] = r_k \boldsymbol{I}_{m_k}, \quad \mathbb{E}[V_k^\top V_k] = n_k \boldsymbol{I}_{r_k}, \quad \mathbb{E}[V_k V_k^\top] = r_k \boldsymbol{I}_{n_k}. \tag{11}$$

These properties hold for independent standard Gaussian or Rademacher entries. Other perturbation constructions [50] require separate verification of the independence and moment assumptions.

Theorem 4.4 (Convergence of the Base Estimator)**.** *Under Asms. 4.1 to 4.3, consider unclipped LR-Q-SPSA updates with fresh factors at every step and constant applied step size $0 < \eta \le 1/(2LC_2')$. Let $m = \max_k m_k$, $n = \max_k n_k$, and let C_1, C_2 be the moment constants in Thm. G.2. Define*

$$A_\mu = C_1 L^2 \mu^2 \mathcal{K}^3 m^3 n^3 r, \tag{12}$$

$$C_1' = 2A_\mu + 4C_2 \mathcal{K} mn\sigma^2, \qquad C_2' = 4C_2 \mathcal{K} mn. \tag{13}$$

If $f^* = \inf_{\mathbf{\Delta}} f(\mathbf{\Delta}) > -\infty$, then for any $T \ge 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(\mathbf{\Delta}_t)\|^2 \le \frac{4[f(\mathbf{\Delta}_0) - f^*]}{\eta T} + 2A_\mu + 2L\eta C_1'. \tag{14}$$

The stated bound separates a finite-difference approximation term, which scales with r for a common rank across blocks, from a coarse projection second-moment term proportional to mn. It does not explicitly assume a low-rank true gradient and therefore does not by itself prove that low-rank perturbations outperform QZO when the true gradient is low-rank. The controlled spectral comparisons in Sec. H.1 provide empirical support for that motivation. The bound offers qualitative information about perturbation moments. Its large dimension-dependent constants do not prescribe practical step sizes or determine the empirically best r and τ. For the mixed perturbations used by LQZO, the geometric and empirical benefits of reshaping are not implied by this bound.

5 Experiments

We evaluate LQZO across architectures, scales, and quantization levels and study its hyperparameters.

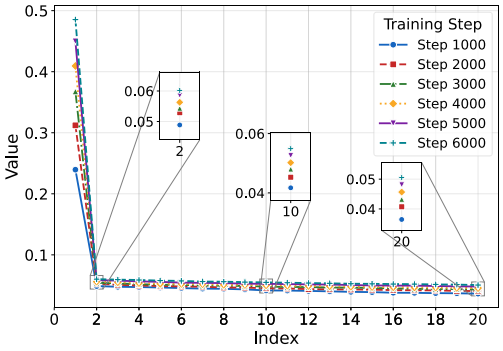


Figure 5: Top 20 of 512 singular values of LQZO checkpoint displacements for the *fc2* module in layer 31 of OPT-6.7B.

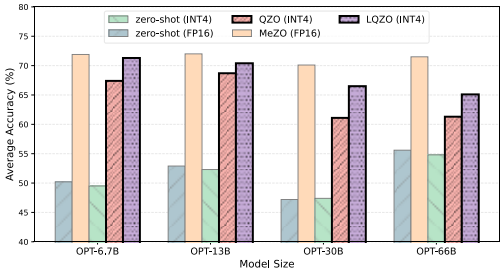


Figure 6: ZO optimizer scores across OPT model scales, averaged over SST-2, RTE, BoolQ, WiC, WSC, and SQuAD (Tabs. 1 and 4).

5.1 Experimental Settings

We evaluate LQZO on six LLMs from the OPT and LLaMA families across 11 benchmarks, as well as ViT on three image classification datasets. The main OPT experiments cover INT8, INT4, and INT2, while the ViT experiments cover INT8 and INT4. Additional experiments evaluate Qwen3-4B-Base at INT4 on five tasks and GPTQ group sizes 32, 64, and 128 on OPT-6.7B. Fine-tuning task scores are averaged over three random seeds, with standard deviations reported in tables. Detailed configurations are provided in Apps. A and H. Unless otherwise specified, ablation studies are conducted on OPT-6.7B using the SST-2 dataset to capture representative performance trends. Zero-shot evaluation separates quantization-induced loss from fine-tuning gains. QZO is the primary quantized scale-optimization baseline, while FP16 MeZO provides a full-precision reference. For the LLM experiments, checkpoints are selected by validation loss rather than by test performance.

5.2 Main Results

Spectral Analysis. Figs. 2, 3 and 5 compare weight gradients from Adam fine-tuning with QZO and LQZO scale displacements during fine-tuning. These gradients exhibit concentrated spectra, with fewer than 40 singular values above 0.1. QZO displacements have a flatter spectrum within a narrow range (< 0.02), whereas LQZO retains a dominant singular value (> 0.2) and smaller values below 0.05. Absolute magnitudes are not directly comparable across these parameterizations. Sec. H.1 complements this analysis with FO scale gradients and QZO/LQZO displacements on the same GPTQ INT4 model. Attention scale gradients and LQZO updates have more concentrated spectra than QZO displacements. Flatter MeZO displacement spectra also occur in the FP16 comparison. These observations support the low-rank motivation but do not establish universal spectral behavior or explain the gains uniquely. In the controlled comparison, output and feed-forward gradients are less concentrated than attention gradients, indicating that spectral structure varies across modules. Moreover, cumulative displacements integrate multiple stochastic updates, so their spectra need not reproduce those of an instantaneous gradient.

Performance across Quantization Levels. Tab. 1 presents the performance of various ZO optimizers when fine-tuning OPT-6.7B across 11 tasks and three quantization levels. LQZO consistently outperforms QZO, with average score improvements of 2.0%, 2.2%, and 3.3% at INT8, INT4, and INT2 precisions, respectively. At INT8 and INT4, LQZO nearly matches the performance of full-precision MeZO, trailing by 0.7% and 0.8%. The INT2 results illustrate the limits imposed by extreme quantization. The zero-shot average score drops to 37.6% (a 14.6% reduction from the FP16 zero-shot score), indicating severe information loss. Despite these challenges, LQZO achieves a 16.7% improvement over the INT2 zero-shot score and outperforms QZO by 3.3%, demonstrating its effectiveness in mitigating the degradation caused by quantization. Fig. 1 shows that LQZO narrows the performance gap on SST-2 but cannot fully recover full-precision performance under extreme quantization.

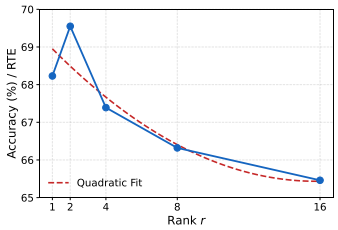


Figure 7: Perturbation rank r.

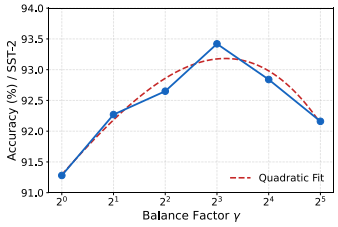


Figure 8: Balance factor γ.

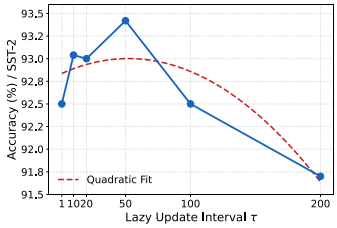


Figure 9: Update interval τ.

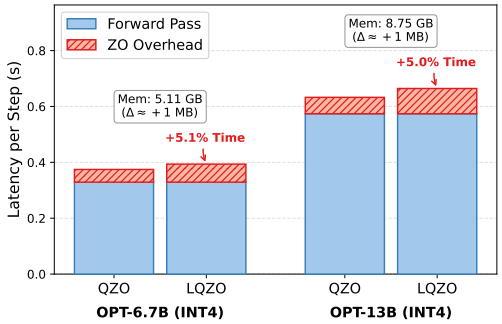


Figure 10: GPU elapsed-time breakdown at INT4. LQZO adds only $\sim$5% overhead over QZO, as the forward pass dominates per-step time.

Generalization across Model Architectures and Scales. Fig. 6 compares average scores on six tasks across different OPT model scales. LQZO consistently outperforms QZO while closely approaching the performance of MeZO, demonstrating scalability across model sizes. Notably, as model size increases, quantization methods generally exhibit performance degradation, whereas MeZO maintains stability. LQZO partially mitigates this scaling challenge, showing more graceful degradation compared to QZO. We further validate the architectural generalization of LQZO on LLaMA-2-7B and LLaMA-3.1-8B at INT4 precision, where LQZO outperforms QZO by 1.6% and 3.8%, respectively, while closely approaching MeZO's performance (see App. C). On Qwen3-4B-Base with GPTQ INT4, LQZO improves the average over five tasks by 1.15% (83.44 to 84.59, Sec. H.3). Across GPTQ group sizes 32, 64, and 128 on OPT-6.7B, the SST-2 gains range from 1.30% to 5.24% (Sec. H.2). On Qwen, gains occur on four of the five tasks, while SST-2 scores remain close to QZO. Varying the GPTQ group size also changes the number and shape of trainable scales, testing the optimizer under different scale granularities.

Generalization to Computer Vision Tasks. To validate LQZO's cross-modality versatility, we fine-tune a ViT-base-patch16-224 with RTN weight quantization on three image classification benchmarks (CIFAR-10, Oxford-IIIT Pet, CIFAR-100). As detailed in App. D, LQZO consistently outperforms QZO across all evaluated bit-widths, reaching a gain of 13.4% on CIFAR-100 at INT4. These results provide evidence beyond GPTQ-quantized LLMs.

Computational and Memory Efficiency Analysis. A critical concern for LQZO is the practical trade-off between performance gains and the computational overhead introduced by low-rank matrix reconstruction. To address this, we measure the peak memory usage and per-step GPU elapsed time on an NVIDIA A100 GPU (Sequence Length = 128, Batch Size = 16).

Quantization consistently yields massive memory reductions compared to full-precision MeZO (e.g., up to 3.60$\times$ at INT2, as detailed in App. E), and LQZO maintains nearly identical peak memory usage to QZO (with a negligible delta of $\sim$1 MB) with the persistent factors $\mathbf{V}$ refreshed lazily and $\mathbf{U}$ regenerated from random seeds. This small auxiliary state preserves most of the memory savings from low-bit weight storage. As illustrated in Fig. 10, the time decomposition reveals that the LLM forward pass is the absolute bottleneck. The measured GPU elapsed time of LQZO is approximately 5.1% higher than that of QZO for OPT-6.7B and 5.0% higher for OPT-13B. These measurements quantify reconstruction costs for this workload. See App. A for the timing protocol. We report mean GPU time over 100 iterations after 10 warmup steps, using CUDA events on the GPU stream.

Timings cover forward passes, perturbation reconstruction, and scale updates. They exclude data loading and host preprocessing and do not measure end-to-end fine-tuning throughput.

5.3 Ablation Study

Influence of the perturbation rank r. The approximation term in Thm. 4.4 grows with r under fixed factor distributions, but upper bounds do not order actual convergence rates or task scores. On OPT-6.7B/RTE, Fig. 7 shows the best measured performance at $r = 2$ and deterioration at larger tested ranks. Rank-one perturbations can restrict exploration within a lazy-update interval, suggesting an empirical balance between exploration and estimation cost. With refreshed subspaces, the selected rank need not match the true gradient rank. The bound prescribes neither rank nor interval.

Influence of the balance factor γ. In Sec. 4.2.3, we derive the optimal balance factor γ that minimizes the sum of dimensions in reshaped matrices (Eq. (7)). We empirically evaluate this by varying γ during fine-tuning of OPT-6.7B on SST-2 using LQZO. Fig. 8 shows that the optimal γ is 2^3. The hidden embedding dimension of OPT-6.7B is 4096, resulting in most matrices having a shape of $\mathbb{R}^{4096\times4096}$. With a group size of 128 in GPTQ, the quantization scale matrices typically have dimensions $\mathbb{R}^{4096\times32}$. For this shape, $\gamma = 8$ and $\gamma = 16$ both minimize the constrained dimension sum, since $4096/8 + 8 \cdot 32 = 4096/16 + 16 \cdot 32 = 768$. The empirical choice $\gamma = 8$ therefore coincides with one geometric minimizer. The criterion does not predict a unique accuracy optimum.

Influence of the lazy update interval τ. The parameter τ controls the refresh frequency of each factor V_k, which defines the row subspace for low-rank perturbations in block k. An appropriate choice of τ balances convergence within the subspace with sufficient exploration across fine-tuning steps. Reusing the factor keeps updates in a shared row subspace during each interval, while refreshing it opens new directions. The interval therefore controls both sampling overhead and the diversity of subspaces visited over a fixed optimization budget. We analyze the impact of τ on accuracy when fine-tuning OPT-6.7B on SST-2 using LQZO. Fig. 9 shows that optimal performance is achieved around $\tau = 50$, suggesting that this lazy update interval provides an effective trade-off between stabilizing subspace directions and exploring new ones.

6 Conclusion

We introduced LQZO for quantized scale optimization, combining low-rank perturbations, Binary-Aggregated Gaussian Perturbation, and Isotropy-Enhanced Estimation. We proved convergence properties of the independently refreshed, unclipped base estimator and evaluated the practical optimizer. LQZO achieves higher average task scores than QZO and approaches FP16 MeZO with lower memory use in the tested settings.

Limitations. Benefits may vary across modules and diffuse gradient spectra. Evaluation covers GPTQ-quantized dense LLMs, three group sizes, and RTN-quantized ViT. Broader quantization schemes, MoE models, and joint weight and activation quantization remain untested. The method requires continuous scales with fixed quantized weights. ZO optimization can converge more slowly than first-order methods, and forward activations and workspaces still require memory. The guarantee excludes lazy updates and clipping and does not determine the best rank or update interval.

Broader Impacts. Lower-memory fine-tuning may broaden access to large-model adaptation. Adapted models can retain biases or enable misuse. Downstream deployments require task-specific evaluation and responsible use.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grants 62476099, 62076101, and 62576364), the Guangdong Basic and Applied Basic Research Foundation (Grants 2024B1515020082 and 2026B1515020071), the Shenzhen Science and Technology Program (Grants JCYJ20241202124430041 and SYSRD20250529113401002), the 2026 Guangdong Provincial Special Fund for Industrial Foundation Reengineering (Multimodal Adaptive Embodied AI Large Model Project), the TCL Young Scholars Program, and the High Performance Computing Platform of South China University of Technology.

References

[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

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

[5] Jun Rao, Fei Wang, Liang Ding, Shuhan Qi, Yibing Zhan, Weifeng Liu, and Dacheng Tao. Where does the performance improvement come from? -a reproducibility concern about image-text retrieval. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2727–2737, 2022.

[6] Fei Wang, Liang Ding, Jun Rao, Ye Liu, Li Shen, and Changxing Ding. Can linguistic knowledge improve multimodal alignment in vision-language pretraining? *arXiv preprint arXiv:2308.12898*, 2023.

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

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

[9] Huanyu Wang, Junjie Liu, Xin Ma, Yang Yong, Zhenhua Chai, and Jianxin Wu. Compressing models with few samples: Mimicking then replacing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 701–710, 2022.

[10] Xinglong Sun, Ali Hassani, Zhangyang Wang, Gao Huang, and Humphrey Shi. Disparse: Disentangled sparsification for multitask model compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12382–12392, 2022.

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

[12] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35: 30318–30332, 2022.

[13] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[14] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pages 38087–38099. PMLR, 2023.

[15] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100, 2024.

[16] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

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

[18] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

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

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

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

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

[23] J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[24] Sifeng Shang, Jiayi Zhou, Chenyu Lin, Minxian Li, and Kaiyang Zhou. Fine-tuning quantized neural networks with zeroth-order optimization. *arXiv preprint arXiv:2505.13430*, 2025.

[25] Noga Bar and Raja Giryes. Zoqo: Zero-order quantized optimization. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[26] Chen Feng, Shaojie Zhuo, Xiaopeng Zhang, Ramchalam K Ramakrishnan, Zhaocong Yuan, and Andrew Z Li. Stepping forward on the last mile. *Advances in Neural Information Processing Systems*, 37:94851–94870, 2024.

[27] Jiajun Zhou, Yifan Yang, Kai Zhen, Ziyue Liu, Yequan Zhao, Ershad Banijamali, Athanasios Mouchtaris, Ngai Wong, and Zheng Zhang. QuZO: Quantized zeroth-order fine-tuning for large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5341–5359, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6.

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

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

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

[31] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

[32] Brett W Larsen, Stanislav Fort, Nic Becker, and Surya Ganguli. How many degrees of freedom do we need to train deep networks: a loss landscape perspective. In *International Conference on Learning Representations*, 2022.

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

[34] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[35] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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, Online, August 2021. Association for Computational Linguistics.

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

[37] Yan Sun, Qixin Zhang, Tiansheng Huang, Liang Ding, Li Shen, and Dacheng Tao. TeZO: Empowering the low-rankness on the temporal dimension in the zeroth-order optimization for fine-tuning LLMs, 2026.

[38] Feihu Jin, Shipeng Cen, and Ying Tan. Prior-informed zeroth-order optimization with adaptive direction alignment for memory-efficient llm fine-tuning. *arXiv preprint arXiv:2601.04710*, 2026.

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

[40] Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Sparse meZO: Less parameters for better performance in zeroth-order LLM fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[41] Ziming Yu, Pan Zhou, Sike Wang, Jia Li, and Hua Huang. Subzero: Random subspace zeroth-order optimization for memory-efficient llm fine-tuning. *arXiv preprint arXiv:2410.08989*, 2024.

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

[43] Yuang Liu, Wei Zhang, and Jun Wang. Zero-shot adversarial quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1512–1521, 2021.

[44] Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. Pd-quant: Post-training quantization based on prediction difference metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24427–24437, 2023.

[45] Jaehyeon Moon, Dohyung Kim, Junyong Cheon, and Bumsub Ham. Instance-aware group quantization for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16132–16141, 2024.

[46] Minghao Fu, Hao Yu, Jie Shao, Junjie Zhou, Ke Zhu, and Jianxin Wu. Quantization without tears. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4462–4472, 2025.

[47] Junhan Kim, Gukryeol Lee, Seungwoo Son, Jeewook Kim, and Yongkweon Jeon. Two-stage grid optimization for group-wise quantization of llms. *arXiv preprint arXiv:2602.02126*, 2026.

[48] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP$\#$: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Forty-first International Conference on Machine Learning*, 2024.

[49] Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *Proceedings of the 41st International Conference on Machine Learning*, pages 12284–12303, 2024.

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

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

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

[53] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[54] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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

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

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

[58] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, 2019.

[59] Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, 2019.

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

[61] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 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, 2018.

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

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

[64] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

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

[66] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[67] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[68] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[69] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

A Experimental Details

A.1 Baselines

We conducted comparative evaluations of LQZO against three baseline approaches: zero-shot, MeZO [19], and QZO [24]. The zero-shot approach evaluates both pre-trained and quantized models without fine-tuning, and provides a reference without fine-tuning. MeZO serves as a full-precision (FP16) reference. QZO [24] serves as our primary baseline for scale-wise quantized ZO optimization across INT8, INT4, and INT2 precisions.

A.2 Models, Datasets, and Metrics

Our main LLM experiments encompass two prominent model families: the OPT [51] family and the LLaMA [52, 4] family. To ensure comprehensive coverage of model scales, we selected four models from the OPT [51] family: OPT-6.7B, OPT-13B, OPT-30B, and OPT-66B. For contemporary relevance, we included LLaMA-2-7B [52] and LLaMA-3.1-8B [4] from the LLaMA family. Additional experiments on Qwen3-4B-Base [53] are reported in Sec. H.3. We evaluate eleven downstream tasks commonly used in ZO optimization. These include seven classification tasks, including tasks from SuperGLUE [54]: SST-2 [55], RTE [56], CB [57], BoolQ [58], WiC [59], WSC [60], and MultiRC [61]. The two multiple-choice tasks from the SuperGLUE benchmark are COPA [62] and ReCoRD [63]. The two question-answering tasks are SQuAD [64] and DROP [65], which we treat as generation tasks. We report accuracy for classification and multiple-choice tasks and F1 for generation tasks.

For computer vision tasks, we evaluate LQZO on the **ViT-base-patch16-224** model [66]. The evaluation spans three datasets: CIFAR-10, CIFAR-100 [67], and Oxford-IIIT Pet [68]. For these tasks, we report top-1 accuracy as the primary metric.

Table 2: Hyperparameter configurations for ZO optimizers in LLM and ViT fine-tuning. Learning rates were selected via grid search over the specified ranges. LQZO rates are base rates before the bit-width scaling in Alg. 1.

Method	Hyperparameter	Value (LLMs)	Value (ViT)
MeZO	Batch Size	16 (8 for DROP)	64
	Learning Rate η	{1e-7, 5e-7, 1e-6}	{1e-5, 3e-5}
	Perturbation Scale μ	1e-3	
QZO	Batch Size	16 (8 for DROP)	64
	Learning Rate η	{5e-8, 1e-7, 5e-7}	{3e-6, 7e-6, 5e-6, 1e-5}
	Perturbation Scale μ	1e-3	
	Clipping Threshold C	100	
LQZO	Batch Size	16 (8 for DROP)	64
	Learning Rate η	{5e-8, 1e-7, 5e-7}	{3e-6, 5e-6, 1e-5, 2e-5}
	Perturbation Scale μ	1e-3	
	Clipping Threshold C	100	
	Rank r	2	
	Interval τ	50	
	Balance Factor γ	8	

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 Tab. 2. For LQZO, we used the default parameters specified in Tab. 2 across all experiments, except in ablation studies where only the parameters under investigation were modified. To maintain exact transparency with our reported figures, exploratory analyses (e.g., Sec. 3) and ablations on the balance factor γ and lazy update interval τ utilize the SST-2 dataset, while the ablation analysis of rank r is conducted on the RTE dataset.

Hardware Configuration. Experiments use NVIDIA A100 GPUs with 40 GB of memory per device. FP16 MeZO runs for OPT-30B and OPT-66B use the minimum feasible GPU count that accommodates model weights and runtime allocations. The efficiency measurements reported below use the stated single-GPU workloads.

Language Modeling Settings: Due to GPU memory constraints (NVIDIA A100-40GB), we reduced the batch size from 16 to 8 for MeZO fine-tuning on OPT-6.7B and LLaMA models when evaluating on the DROP dataset. We employed GPTQ [13] for weight quantization, using a group size of 128 and the *pile-val-backup* dataset[5] [69] for calibration. Since the quantization scaling factors in GPTQ vary with precision levels, we proportionally adjusted the learning rates of QZO and LQZO accordingly, as detailed in Alg. 1. For each factor V_k in LQZO, we sampled independent Bernoulli variables $B_{k,ij} \sim \text{Bernoulli}(0.5)$ and mapped them to $[V_k]_{ij} = 2B_{k,ij} - 1$, yielding the Rademacher distribution on $\{-1, +1\}$. All LLMs were fine-tuned for 20,000 steps with checkpoints saved every 4,000 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 reported the average performance metric along with the standard deviation.

Computer Vision Settings: The ViT models use RTN weight quantization and were fine-tuned using the same optimization logic. To accommodate varying task complexities, models were fine-tuned for a maximum of 20,000 steps on CIFAR-10 and Oxford-IIIT Pet, and extended to 200,000 steps for the more challenging CIFAR-100 dataset. To ensure rigorous evaluation, we used 45,000 training images, 1,000 validation images, and 10,000 test images for CIFAR-10/100. The Oxford-IIIT Pet dataset is split into 3,312 training, 368 validation, and 3,669 testing images. Optimal checkpoints were selected based on validation accuracy. As with the LLM experiments, all results are averaged over three random seeds with standard deviations reported.

GPU Timing Protocol. We measured per-step GPU elapsed time on a single **NVIDIA A100-40GB GPU** using `torch.cuda.Event`, covering the forward passes and the optimization operations, including matrix reconstruction and reshaping. We report the mean over 100 timed iterations after 10 warmup steps. CUDA-event timing measures elapsed time on the GPU stream. It is not an end-to-end measurement of data loading, host-side preprocessing, and training execution.

B Loss Convergence Analysis

To further demonstrate the performance advantages of LQZO in quantized ZO optimization, we conduct a comprehensive analysis of the convergence behavior between QZO and LQZO through their loss trajectories across multiple model scales and tasks.

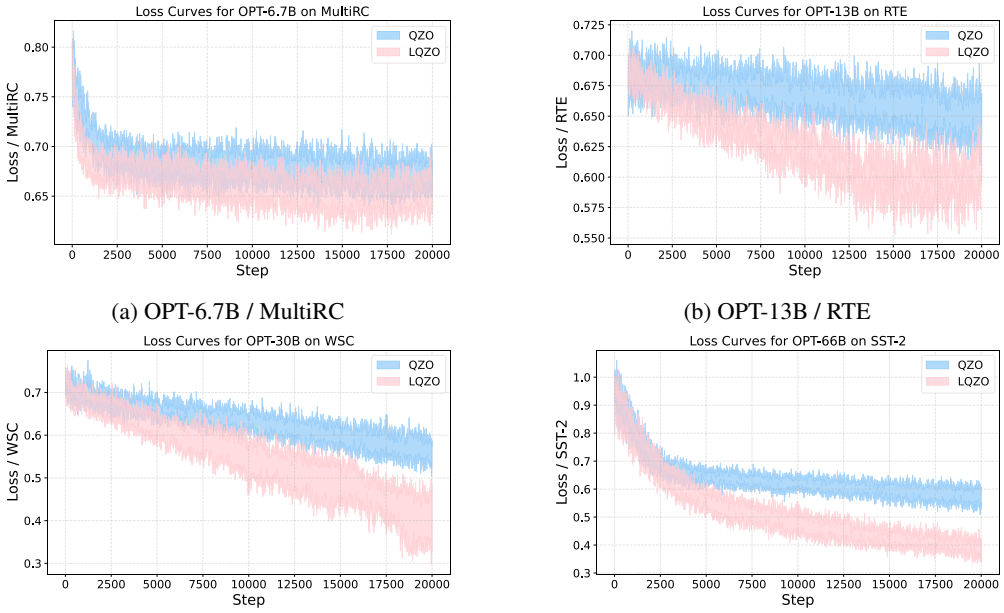
QZO uses dense random perturbations, whereas LQZO uses factorized perturbations $Z_k = U_k V_k^\top$ with rank at most r. Increasing r removes the rank restriction when $r \geq \min(m_k, n_k)$, but does not make the product distribution identical to QZO's Gaussian perturbation distribution. Accordingly, QZO is a related dense-perturbation baseline rather than an exact special case of the factorized estimator. The convergence analysis in Thm. 4.4 concerns independently refreshed, unclipped updates and does not compare the convergence rates of QZO and the practical optimizer with lazy updates and clipping. The loss curves provide an empirical comparison without isolating the cause of the observed gains.

We present the loss convergence curves for OPT models of varying scales (6.7B, 13B, 30B, 66B) across multiple tasks (MultiRC, RTE, WSC, SST-2) in Fig. 11. The experimental results consistently demonstrate that LQZO achieves significantly faster convergence and lower final loss values compared to QZO across all model sizes and tasks.

C Generalization Across Model Architectures

To further validate the architectural generalization of LQZO, we evaluate on LLaMA family models at INT4 precision. As shown in Tab. 3, LQZO outperforms QZO by 1.6% on LLaMA-2-7B and

[5] `https://huggingface.co/datasets/mit-han-lab/pile-val-backup`



(a) OPT-6.7B / MultiRC

(b) OPT-13B / RTE

(c) OPT-30B / WSC

(d) OPT-66B / SST-2

Figure 11: Comparative convergence behavior of training loss for OPT models fine-tuned with QZO and LQZO, showing consistent optimization performance across model sizes and tasks.

Table 3: Entries with $\pm$ report the mean and standard deviation over three random seeds. Performance comparison of ZO optimizers for fine-tuning LLaMA family models at INT4 quantization. Numbers in parentheses represent the absolute improvement of LQZO over QZO.

Model	Task / Method	SST-2	RTE	CB	BoolQ	WiC	WSC	MultiRC	COPA	ReCoRD	SQuAD	DROP	Avg.
		——classification——							–multiple choice–		–generation–		
LLaMA-2 (7B)	zero-shot (FP16)	58.0	61.7	32.1	65.8	50.2	36.5	45.3	79.0	80.7	55.6	19.9	53.2
	MeZO (FP16)	93.7±1.2	64.5±1.8	67.3±1.0	75.3±4.7	62.3±2.9	60.9±2.8	74.3±2.9	85.0±1.7	82.1±0.9	88.2±1.6	41.4±4.3	72.3±2.4
	zero-shot	58.1	59.2	42.9	62.8	50.3	38.5	45.8	82.0	79.0	52.8	18.2	53.6
	QZO	89.8±0.8	61.2±0.8	67.9±0.0	71.9±0.8	60.2±1.1	55.5±10.6	68.6±2.4	84.0±1.0	79.9±0.7	86.5±0.2	42.2±2.7	69.8±1.9
	LQZO	92.7±0.5	68.0±0.8	67.3±1.0	74.8±1.1	62.1±2.6	55.4±8.9	72.3±2.6	85.7±2.1	80.6±1.9	87.9±0.5	38.6±1.3	71.4±2.1 (↑1.6)
LLaMA-3.1 (8B)	zero-shot (FP16)	59.6	45.9	46.4	66.1	52.4	57.7	63.5	85.0	84.2	64.8	27.9	59.4
	MeZO (FP16)	91.9±0.1	65.5±1.9	70.2±1.0	80.5±0.5	56.5±2.7	57.4±8.9	71.6±3.5	89.7±0.6	84.1±0.9	89.7±0.8	63.3±1.3	74.6±2.0
	zero-shot	61.5	46.2	44.6	64.3	50.8	63.5	58.1	87.0	82.8	61.6	23.6	58.5
	QZO	93.6±0.2	62.7±2.5	67.9±1.8	75.0±3.2	56.2±4.3	49.4±7.3	58.8±1.7	90.3±0.6	82.9±0.1	88.7±1.1	47.6±2.4	70.3±2.3
	LQZO	93.2±0.2	68.0±3.2	70.8±1.0	81.2±2.4	58.5±2.5	51.3±7.4	71.9±2.6	90.3±1.5	83.3±0.9	89.1±0.5	57.0±1.8	74.1±2.2 (↑3.8)

3.8% on LLaMA-3.1-8B, approaching MeZO's performance with differences of 0.9% and 0.5%, respectively. Interestingly, although the quantized zero-shot performance of LLaMA-3.1-8B is 0.9% lower than that of the FP16 pre-trained model, this gap narrows to 0.5% following fine-tuning with both MeZO and LQZO, suggesting that our method effectively aids in recovering performance loss induced by quantization.

D Generalization to Computer Vision Tasks

To evaluate the cross-modality generalizability of LQZO, we extend our experiments to Computer Vision by fine-tuning a quantized Vision Transformer (ViT-base-patch16-224). We employ three image classification benchmarks with varying degrees of complexity: CIFAR-10, Oxford-IIIT Pet, and CIFAR-100. Detailed dataset splits and training protocols are provided in App. A.

As summarized in Tab. 5, LQZO consistently and significantly outperforms QZO across all bit-widths. Crucially, the performance gap between LQZO and QZO widens substantially as the task complexity increases. On the relatively simpler CIFAR-10 dataset, LQZO yields an approximately 2% improvement over QZO at INT4. However, on the much more challenging CIFAR-100 dataset, LQZO achieves a 13.4% absolute accuracy gain over QZO at INT4 (51.2% vs. 37.8%), and an approximately 13% gain at INT8. These results demonstrate improvements on the evaluated vision

Table 4: Entries with ± report the mean and standard deviation over three random seeds. Performance comparison of ZO optimizers for fine-tuning OPT models of varying scales at INT4 quantization. Parentheses show absolute average-score gains over QZO.

	Task	SST-2	RTE	BoolQ	WiC	WSC	SQuAD	Avg.
OPT-13B	zero-shot (FP16)	58.8	59.6	59.0	55.0	38.5	46.2	52.9
	MeZO (FP16)	91.4±0.3	69.1±0.9	72.8±1.4	58.0±4.5	55.8±9.2	84.9±0.8	72.0±2.8
	zero-shot	58.0	58.8	57.5	54.1	38.5	47.0	52.3
	QZO	90.2±0.3	62.2±0.9	66.6±0.4	56.8±1.1	51.6±7.2	84.5±0.3	68.7±1.7
	LQZO	90.3±0.8	65.2±4.3	67.8±1.5	58.5±2.5	57.1±8.6	83.2±0.9	70.4±3.1 (↑1.7)
OPT-30B	zero-shot (FP16)	56.7	52.0	39.1	38.5	50.2	46.6	47.2
	MeZO (FP16)	89.9±0.3	66.2±2.7	68.6±0.9	56.6±1.9	58.7±7.5	85.4±0.9	70.9±2.4
	zero-shot	56.5	51.3	38.7	41.4	49.7	46.8	47.4
	QZO	80.5±1.6	56.9±3.8	59.4±5.6	50.8±0.5	40.4±5.0	78.7±1.4	61.1±3.0
	LQZO	86.9±1.1	60.9±0.8	65.8±0.1	55.8±0.6	56.1±6.8	77.5±1.7	67.1±1.8 (↑6.0)
OPT-66B	zero-shot (FP16)	57.5	67.2	66.8	50.6	43.3	48.1	55.6
	MeZO (FP16)	91.1±0.5	69.4±1.5	68.6±0.9	56.2±1.7	58.3±8.1	85.4±0.9	71.5±2.3
	zero-shot	56.9	65.7	66.0	52.4	38.5	49.7	54.8
	QZO	65.9±1.6	52.8±2.3	64.1±0.4	55.3±1.8	53.8±15.0	75.7±1.4	61.3±3.8
	LQZO	83.7±0.5	50.9±1.5	67.6±1.8	51.6±0.8	58.7±7.5	78.0±1.1	65.1±2.2 (↑3.8)

Table 5: Entries with ± report the mean and standard deviation over three random seeds. Performance comparison of ZO optimizers for fine-tuning Vision Transformers (ViT-base-patch16-224) across different quantization precisions. Parentheses show absolute average-score gains over QZO.

Precision	**Method**	**CIFAR-10**	**Oxford-IIIT-Pet**	**CIFAR-100**	**Avg.**
FP16	zero-shot	11.2	3.6	1.1	5.3
	MeZO	94.4±0.3	83.5±0.2	66.1±0.6	81.3±0.4
INT8	zero-shot	11.2	3.5	1.1	5.3
	QZO	92.0±0.7	68.5±1.3	35.6±2.8	65.4±1.6
	LQZO	93.9±0.2	74.4±0.3	48.6±0.8	72.3±0.4 (↑6.9)
INT4	zero-shot	12.3	2.9	1.1	5.4
	QZO	91.4±0.7	68.5±2.8	37.8±2.8	65.9±2.1
	LQZO	93.5±0.2	76.6±1.6	51.2±1.4	73.7±1.0 (↑7.8)

tasks, with larger gains on CIFAR-100. They extend the empirical evidence beyond LLMs. Task accuracy alone does not identify the gradient structure behind these gains.

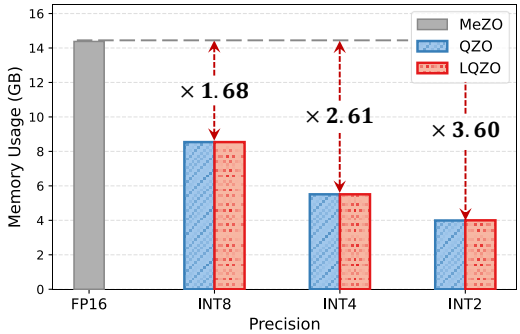


Figure 12: Memory usage for fine-tuning OPT-6.7B with different ZO optimizers across various quantization precisions, using an input length of 128 and a batch size of 16.

Table 6: Entries with ± report the mean and standard deviation over three random seeds. Performance comparison of *standard Gaussian* and *Rademacher* sampling for V_k when fine-tuning OPT-6.7B. The red numbers in parentheses indicate the performance improvement obtained by using Rademacher perturbations relative to standard Gaussian perturbations.

	SST-2	RTE	BoolQ	WiC	WSC	SQuAD	Avg.
QZO	87.7±0.6	62.3±2.1	64.9±1.7	57.1±1.8	53.5±12.2	79.1±1.2	67.4±3.3
LQZO-Gaussian	92.4±0.8	67.5±1.3	66.4±2.2	60.2±3.2	56.7±7.5	79.9±1.1	70.5±2.7 (↑**3.1**)
LQZO-Rademacher	92.9±1.0	69.6±1.8	67.3±1.0	61.4±1.4	57.1±10.3	79.6±0.5	71.3±2.7 (↑**3.9**)

E Memory Efficiency Analysis

Fig. 12 compares peak memory usage across ZO optimizers when fine-tuning OPT-6.7B at various quantization precisions (Sequence Length = 128, Batch Size = 16). Following the simulation protocol of Wang et al. [21], we measured peak tensor memory allocated through the PyTorch CUDA allocator using `torch.cuda.max_memory_allocated()`.

Quantization consistently yields substantial memory reductions compared to full-precision MeZO. Specifically, INT8 quantization reduces memory by $1.68\times$, INT4 by $2.61\times$, and INT2 achieves the largest reduction of $3.60\times$, demonstrating the effectiveness of weight quantization in alleviating the memory bottleneck of LLM fine-tuning.

Crucially, LQZO maintains nearly identical peak memory to QZO across all precision levels, with a negligible delta of $\sim$1 MB. The persistent auxiliary state consists of the factors $\mathbf{V}$ and random seeds. The factors $\mathbf{U}$ are regenerated from their seeds instead of being retained between loss evaluations. Temporary factors, reconstructed perturbations, and matrix-operation workspaces can still contribute to peak memory. The reported difference is an empirical measurement under the specified hardware and workload. Lazy updates amortize factor generation by reusing $\mathbf{V}$ for τ steps.

These measurements show that the accuracy gains of LQZO over QZO are obtained with a small additional peak-memory cost in the evaluated setting.

F Ablation on Perturbation Sampling for V_k

To evaluate the effectiveness of our Binary-Aggregated Gaussian Perturbation strategy, we conduct an ablation study by comparing the performance when the factor V_k is sampled from either a standard Gaussian distribution or a Rademacher distribution. As shown in Tab. 6, sampling V_k from the Rademacher distribution results in improved performance on five out of six datasets, although there is a 0.3% decrease in SQuAD F1 relative to the Gaussian variant. Importantly, even when V_k is sampled from the standard Gaussian distribution, LQZO achieves an average score improvement of 3.1% over QZO. This ablation shows that factorized perturbations improve the average score even with Gaussian V_k, while Rademacher V_k provides an additional average improvement in the evaluated setting. The moment analysis supports the sampling choice, but does not by itself attribute every task-score difference to estimator variance.

G Proof

All results below concern the raw LR-Q-SPSA estimator with independently sampled factors at a fixed parameter value. The second-moment and convergence bounds use a common rank r for every block. The convergence analysis additionally requires fresh factors at every step and excludes scalar clipping, as stated in Asm. 4.3 and Thm. 4.4.

G.1 Asymptotic Unbiasedness of LR-Q-SPSA

Theorem G.1 (Asymptotic Unbiasedness)**.** *Under Asms. 4.1 and 4.3, for a fixed* $\mathbf{\Delta}$ *and mini-batch* $\mathcal{B}$*,*

$$\lim_{\mu \to 0} \mathbb{E}_{\mathbf{U},\mathbf{V}}[\widehat{\nabla}\mathcal{L}(\mathbf{\Delta};\mathcal{B})] = \nabla\mathcal{L}(\mathbf{\Delta};\mathcal{B}). \tag{15}$$

Algorithm 1: Low-rank Quantized ZO Optimizer (LQZO)

Input: stored quantization scales $\mathbf{\Delta}^{\text{orig}}$, scale-space loss $\mathcal{L}$, step budget T, training dataset $\mathcal{D}$, perturbation scale μ, learning-rate schedule $\{\eta_t\}$, refresh interval τ, positive rank r, bit width b, compatible reshape factor γ, and clipping threshold C.

Set $m_k = M_k/\gamma$, $n_k = \gamma N_k$, and $r_k = r$ for each stored block of shape $M_k \times N_k$

$\mathbf{\Delta} \leftarrow \mathcal{R}(\mathbf{\Delta}^{\text{orig}})$ `// Work in perturbation coordinates`

for $t = 0, \dots, T-1$ **do**

  Sample mini-batch $\mathcal{B}_t$ from $\mathcal{D}$ and random seed s

  $\eta_t^{\text{eff}} \leftarrow 2^{4-b}\eta_t$ `// Scaling according to quantization precision`

  foreach $k = 1, \dots, \mathcal{K}$ **do**

    if $t \bmod \tau = 0$ **then**

      Sample $V_k \in \mathbb{R}^{n_k \times r_k}$ with independent Rademacher entries `// Resample` V_k

  $\mathbf{\Delta} \leftarrow \text{Perturbation}(\mathbf{\Delta}, \mu, \mathbf{V}, s)$

  $l_+ \leftarrow \mathcal{L}(\mathbf{\Delta}; \mathcal{B}_t)$

  $\mathbf{\Delta} \leftarrow \text{Perturbation}(\mathbf{\Delta}, -2\mu, \mathbf{V}, s)$

  $l_- \leftarrow \mathcal{L}(\mathbf{\Delta}; \mathcal{B}_t)$

  $\mathbf{\Delta} \leftarrow \text{Perturbation}(\mathbf{\Delta}, \mu, \mathbf{V}, s)$ `// Restore the unperturbed scales`

  $g_{\text{est},t} \leftarrow (l_+ - l_-)/(2\mu)$ `// Calculate finite difference`

  $g_{\text{clip},t} \leftarrow \max(-C, \min(C, g_{\text{est},t}))$ `// Clip values`

  Reset the random number generator with seed s `// Replay the same` U_k `sequence`

  foreach $k = 1, \dots, \mathcal{K}$ **do**

    Sample $U_k \in \mathbb{R}^{m_k \times r_k}$ with independent standard Gaussian entries

    $\Delta_k \leftarrow \Delta_k - \eta_t^{\text{eff}} g_{\text{clip},t} U_k V_k^\top / r_k$ `// Update quantization scale in place`

return $\mathbf{\Delta}^{\text{orig}} = \mathcal{R}^{-1}(\mathbf{\Delta})$

Function `Perturbation(`$\mathbf{\Delta}, \alpha, \mathbf{V}, s$`):`

  Reset the random number generator with seed s

  foreach $k = 1, \dots, \mathcal{K}$ **do**

    Sample $U_k \in \mathbb{R}^{m_k \times r_k}$ with independent standard Gaussian entries

    $\Delta_k \leftarrow \Delta_k + \alpha U_k V_k^\top$ `// Modify quantization scale in place`

  return $\mathbf{\Delta}$

Proof. Write $\mathbf{G} := \nabla\mathcal{L}(\mathbf{\Delta}; \mathcal{B}) = \{G_k\}_{k=1}^{\mathcal{K}}$, with $G_k = \nabla_{\Delta_k}\mathcal{L}(\mathbf{\Delta}; \mathcal{B})$, and let $\mathcal{P} := \{\mathcal{P}_k\}_{k=1}^{\mathcal{K}}$ be the linearized estimator

$$\mathcal{P}_k = \frac{1}{r_k}\left(\sum_{l=1}^{\mathcal{K}} \langle G_l, Z_l \rangle\right) Z_k. \tag{16}$$

Independence and the unit second moments give $\mathbb{E}[[Z_k]_{ij}[Z_k]_{pq}] = r_k\delta_{ip}\delta_{jq}$, where δ_{ip} is the Kronecker delta. The mean of each Z_k is zero, so cross-block terms vanish in the first moment. Therefore $\mathbb{E}[\mathcal{P}_k] = G_k$. Let $g_{\text{est}} = [\mathcal{L}(\mathbf{\Delta} + \mu\mathbf{Z}; \mathcal{B}) - \mathcal{L}(\mathbf{\Delta} - \mu\mathbf{Z}; \mathcal{B})]/(2\mu)$. By L-smoothness,

$$|g_{\text{est}} - \langle \mathbf{G}, \mathbf{Z} \rangle| \le L\mu\|\mathbf{Z}\|^2. \tag{17}$$

Consequently, $\mathbb{E}\|\widehat{\nabla}\mathcal{L} - \mathcal{P}\|^2 \le L^2\mu^2\mathbb{E}\|\mathbf{Z}\|^6/r_{\min}^2$, where $r_{\min} = \min_k r_k$. The sixth moment is finite. Jensen's inequality then shows that the estimator's mean approaches $\mathbf{G}$ as $\mu \to 0$. This establishes asymptotic unbiasedness without claiming exact unbiasedness for finite μ. □

G.2 Second Moment of LR-Q-SPSA

Theorem G.2 (Second-Moment Bound)**.** *Under Asms. 4.1 to 4.3, at a fixed $\mathbf{\Delta}$ with common rank r, the expectations below average over the independently sampled mini-batch and both factors:*

$$\mathbb{E}_{\mathcal{B},\mathbf{U},\mathbf{V}}\|\widehat{\nabla}\mathcal{L}(\mathbf{\Delta}; \mathcal{B})\|^2 \le 2A_\mu + 4C_2\mathcal{K}mn\big(\|\nabla f(\mathbf{\Delta})\|^2 + \sigma^2\big), \tag{18}$$

where $A_\mu = C_1L^2\mu^2\mathcal{K}^3m^3n^3r$ and C_1, C_2 are finite moment-dependent constants. Moreover,

$$\|\mathbb{E}_{\mathcal{B},\mathbf{U},\mathbf{V}}[\widehat{\nabla}\mathcal{L}(\mathbf{\Delta}; \mathcal{B})] - \nabla f(\mathbf{\Delta})\|^2 \le A_\mu. \tag{19}$$

Proof. We separate the finite-difference error from the linearized estimator:

$$\mathbb{E}\|\widehat{\nabla}\mathcal{L}\|^2 \le 2\mathbb{E}\|\widehat{\nabla}\mathcal{L} - \mathcal{P}\|^2 + 2\mathbb{E}\|\mathcal{P}\|^2. \tag{20}$$

Finite-difference error. For a block with $d_k = m_k n_k$ entries, the inequality $(\sum_{i,j}[Z_k]_{ij}^2)^3 \le d_k^2 \sum_{i,j}[Z_k]_{ij}^6$ and independence across the r summands in each $[Z_k]_{ij}$ imply $\mathbb{E}\|Z_k\|^6 \le C_1 d_k^3 r^3$. For fixed factor distributions, C_1 is independent of m_k, n_k, r. The product summands have $\nu_4 = \mathbb{E}[[U_k]_{ij}^4]\mathbb{E}[[V_k]_{ij}^4]$ and $\nu_6 = \mathbb{E}[[U_k]_{ij}^6]\mathbb{E}[[V_k]_{ij}^6]$. For independent symmetric zero-mean, unit-variance summands, the sixth moment of their sum is $r\nu_6 + 15r(r-1)\nu_4 + 15r(r-1)(r-2)$, which is bounded by $(\nu_6 + 15\nu_4 + 15)r^3$. Choose C_1 to dominate this coefficient uniformly across blocks. Applying the same inequality across blocks gives

$$\mathbb{E}\|\mathbf{Z}\|^6 \le C_1 \mathcal{K}^3 m^3 n^3 r^3. \tag{21}$$

The common-rank normalization and Eq. (17) yield

$$\mathbb{E}\|\widehat{\nabla}\mathcal{L} - \mathcal{P}\|^2 \le \frac{L^2\mu^2}{r^2}\mathbb{E}\|\mathbf{Z}\|^6 \le A_\mu. \tag{22}$$

Since $\mathbb{E}[\mathcal{P}] = \nabla f(\mathbf{\Delta})$ after averaging over the mini-batch, Jensen's inequality also proves Eq. (19).

Single-block second moment. For the fixed block gradient $G_k \in \mathbb{R}^{m_k \times n_k}$, put $\kappa_4^U = \mathbb{E}[[U_k]_{ij}^4]$ and $\kappa_4^V = \mathbb{E}[[V_k]_{ij}^4]$. Expanding $Z_k = \sum_{a=1}^r u_{k,a} v_{k,a}^\top$, where $u_{k,a}, v_{k,a}$ are columns of U_k, V_k, and grouping the four rank indices gives

$$\begin{aligned} T_k &:= \mathbb{E}[\langle G_k, Z_k\rangle^2 \|Z_k\|^2] \\ &= \|G_k\|^2 \left[r(\kappa_4^U + m_k - 1)(\kappa_4^V + n_k - 1) + r(r-1)(m_k n_k + 2)\right]. \end{aligned} \tag{23}$$

To verify the expansion, all four equal rank indices contribute the first term. The pairing of the two inner-product factors and the two norm factors with distinct rank indices contributes $r(r-1)m_k n_k \|G_k\|^2$. The remaining two pairings each contribute $r(r-1)\|G_k\|^2$. Other patterns contain an independent, zero-mean factor and vanish. Since $\kappa_4^U, \kappa_4^V \ge 1$, $(\kappa_4^U + m_k - 1)(\kappa_4^V + n_k - 1) \le \kappa_4^U \kappa_4^V m_k n_k$ and $m_k n_k + 2 \le 3 m_k n_k$. Thus, for $C_2 \ge \max\{\kappa_4^U \kappa_4^V, 3\}$ uniformly across blocks,

$$T_k / r^2 \le C_2 m_k n_k \|G_k\|^2. \tag{24}$$

For standard Gaussian U_k, choosing Rademacher rather than Gaussian V_k reduces the exact single-block second moment through $\kappa_4^V = 1$ rather than 3. It also reduces the fourth and sixth moments entering the finite-difference bound. For mixed Gaussian and Rademacher factors, $[Z_k]_{ij}/r \sim \mathcal{N}(0, 1/r)$, so

$$\mathbb{E}([Z_k]_{ij}/r)^4 = 3/r^2, \qquad \mathbb{E}([Z_k]_{ij}/r)^6 = 15/r^3. \tag{25}$$

For two Gaussian factors, the corresponding moments are

$$\mathbb{E}([Z_k]_{ij}/r)^4 = 3/r^2 + 6/r^3, \qquad \mathbb{E}([Z_k]_{ij}/r)^6 = 15/r^3 + 90/r^4 + 120/r^5. \tag{26}$$

These expressions follow by expanding the sums of independent product variables with second moment 1 and fourth/sixth moments $(3, 15)$ or $(9, 225)$, respectively. Thus the reduction holds for each positive rank, without asserting joint Gaussianity of all entries of Z_k.

Cross-block terms. The scalar finite difference uses every block, so Eq. (16) cannot be replaced by a block-only inner product. Conditional on the mini-batch, let $a_l = \langle G_l, Z_l\rangle$. Independence yields

$$\begin{aligned} \mathbb{E}\|\mathcal{P}_k\|^2 &= \frac{1}{r^2}\mathbb{E}[a_k^2\|Z_k\|^2] + \frac{1}{r^2}\sum_{l\ne k}\mathbb{E}[a_l^2]\mathbb{E}\|Z_k\|^2 \\ &= T_k/r^2 + d_k \sum_{l\ne k}\|G_l\|^2, \end{aligned} \tag{27}$$

using $\mathbb{E}[a_l^2] = r\|G_l\|^2$ and $\mathbb{E}\|Z_k\|^2 = r d_k$. Summing over k and applying Eq. (24),

$$\mathbb{E}\|\mathcal{P}\|^2 \le \sum_l \left(C_2 d_l + \sum_{k\ne l} d_k\right)\|G_l\|^2 \le C_2 \mathcal{K} mn \|\mathbf{G}\|^2. \tag{28}$$

Finally, Asm. 4.2 gives $\mathbb{E}_\mathcal{B}\|\mathbf{G}\|^2 \le \|\nabla f(\mathbf{\Delta})\|^2 + \sigma^2$. Substituting into Eq. (20) proves the stated, slightly looser bound with coefficient $4C_2\mathcal{K}mn$. □

G.3 Convergence with Finite-Difference Bias

Proof of Thm. 4.4. Let $\mathcal{F}_t$ denote the history before drawing the fresh mini-batch and factors at step t. For $t = 0, \dots, T-1$, put $\widehat{\boldsymbol{g}}_t := \widehat{\nabla}\mathcal{L}(\boldsymbol{\Delta}_t; \mathcal{B}_t)$ and $\boldsymbol{b}_t := \mathbb{E}[\widehat{\boldsymbol{g}}_t \mid \mathcal{F}_t] - \nabla f(\boldsymbol{\Delta}_t)$. By Thm. G.2, $\|\boldsymbol{b}_t\|^2 \le A_\mu$ and $\mathbb{E}[\|\widehat{\boldsymbol{g}}_t\|^2 \mid \mathcal{F}_t] \le C_2'\|\nabla f(\boldsymbol{\Delta}_t)\|^2 + C_1'$. Smoothness of f and the unclipped update give

$$\mathbb{E}[f(\boldsymbol{\Delta}_{t+1}) \mid \mathcal{F}_t] \le f(\boldsymbol{\Delta}_t) - \eta\|\nabla f(\boldsymbol{\Delta}_t)\|^2 - \eta\langle \nabla f(\boldsymbol{\Delta}_t), \boldsymbol{b}_t\rangle + \frac{L\eta^2}{2}\big(C_2'\|\nabla f(\boldsymbol{\Delta}_t)\|^2 + C_1'\big). \tag{29}$$

Using $|\langle \nabla f(\boldsymbol{\Delta}_t), \boldsymbol{b}_t\rangle| \le (\|\nabla f(\boldsymbol{\Delta}_t)\|^2 + \|\boldsymbol{b}_t\|^2)/2$ and $\eta \le 1/(2LC_2')$,

$$\frac{\eta}{4}\|\nabla f(\boldsymbol{\Delta}_t)\|^2 \le f(\boldsymbol{\Delta}_t) - \mathbb{E}[f(\boldsymbol{\Delta}_{t+1}) \mid \mathcal{F}_t] + \frac{\eta A_\mu}{2} + \frac{L\eta^2 C_1'}{2}. \tag{30}$$

Taking total expectations, summing over t, and using $f(\boldsymbol{\Delta}_T) \ge f^*$ yields Eq. (14). □

For fixed μ and η, the bound retains the error floor $2A_\mu + 2L\eta C_1'$ and thus establishes approximate stationarity. For a fixed model, rank, and factor distributions, a family of T-step runs with $\mu_T = c_\mu T^{-1/4}$ and $\eta_T = c_\eta T^{-1/2}$, where $c_\mu > 0$ and $0 < c_\eta \le 1/(2LC_2')$, gives an $O(T^{-1/2})$ bound on the average expected squared gradient norm. These are budget-dependent constant parameters within each run. This observation does not establish convergence of the practical lazy, clipped variant.

G.4 Scope of Lazy Updates and Clipping

The unconditional first-moment calculation requires averaging over both factors at a parameter value independent of those factors. With V_k held fixed during a lazy-update interval, averaging only over fresh U_k gives, in the limit $\mu \to 0$,

$$\lim_{\mu\to 0} \mathbb{E}_{\mathbf{U}}[[\widehat{\nabla}\mathcal{L}(\boldsymbol{\Delta}; \mathcal{B})]_k \mid \boldsymbol{\Delta}, \mathbf{V}, \mathcal{B}] = G_k V_k V_k^\top / r. \tag{31}$$

This is generally not G_k, and the evolving parameters depend on the reused V_k. Scalar clipping further changes the estimator's first moment. Therefore the base-estimator guarantee cannot be transferred directly to the practical lazy, clipped algorithm. Its behavior is assessed by the rank, interval, and sampling ablations, while a full convergence analysis of those modifications remains outside the present theoretical result.

H Additional Controlled Experiments and Robustness Evaluation

H.1 Spectral Comparison on the Same Quantization Scale Parameters

The full-precision weight-gradient and quantized ZO update spectra in Sec. 3 involve different parameterizations. To reduce this confounding factor, we additionally examine exact first-order (FO) scale gradients and the cumulative scale displacements produced by QZO and LQZO on the same quantized model. We use OPT-6.7B with GPTQ INT4, group size 128, SST-2, layer 31, and reshaping factor $\gamma = 8$. For FO, we differentiate the loss with respect to the continuous scales while keeping the integer weights fixed. Backpropagation through $W_{\mathrm{dq}} = \mathcal{R}_k^{-1}(\Delta_k) \odot W_{\mathrm{int}}$ is exact and requires no straight-through estimator. Here W_{int} and W_{dq} denote the fixed quantized and corresponding dequantized weights of block k, respectively. Spectral concentration is measured by σ_1/σ_5 and σ_1/σ_{10}, where σ_i is the ith largest singular value. These ratios normalize out overall magnitude.

Tab. 7 shows that the attention scale gradients have more concentrated spectra than the QZO scale displacements, even when the parameterization is held fixed. The concentration is weaker in the output and feed-forward modules, so the evidence does not support a uniformly low-rank gradient assumption for every block. LQZO produces more concentrated cumulative displacements throughout the measured modules. Because structured perturbations themselves influence displacement rank, these observations motivate the method but do not establish alignment with the true gradient.

We also examine full-precision FP16 OPT-6.7B at layer 31, comparing FO weight gradients with MeZO cumulative weight displacements after 6000 steps. Tab. 8 shows that flatter ZO displacement

Table 7: Spectral ratios on the same quantization scale parameters. Each entry is σ_1/σ_5 (σ_1/σ_{10}). FO measures instantaneous gradients. QZO and LQZO measure cumulative scale displacements at the listed training steps. Each method follows its own training trajectory, so these measurements complement one another rather than compare gradients at a common checkpoint.

Method	Step	k_proj	v_proj	q_proj	out_proj	fc1	fc2
FO	20	3.02 (4.51)	2.68 (3.24)	2.39 (3.25)	1.17 (1.88)	1.14 (1.49)	1.11 (1.30)
FO	60	3.06 (4.42)	3.46 (4.37)	3.78 (5.72)	1.20 (1.84)	1.80 (2.16)	1.09 (1.23)
FO	120	2.94 (4.09)	3.10 (3.83)	4.94 (5.97)	1.31 (2.31)	1.17 (1.73)	1.10 (1.29)
QZO	1000	1.03 (1.07)	1.04 (1.07)	1.04 (1.09)	1.03 (1.07)	1.03 (1.04)	1.03 (1.05)
QZO	3000	1.04 (1.08)	1.03 (1.08)	1.04 (1.09)	1.04 (1.07)	1.02 (1.04)	1.02 (1.05)
QZO	6000	1.03 (1.06)	1.04 (1.06)	1.05 (1.08)	1.06 (1.09)	1.02 (1.04)	1.03 (1.05)
LQZO	1000	4.95 (5.67)	5.49 (6.09)	4.98 (5.80)	4.40 (5.06)	4.76 (5.20)	5.05 (5.50)
LQZO	3000	6.23 (6.78)	7.10 (7.98)	6.16 (6.62)	6.19 (6.63)	6.90 (7.36)	7.21 (7.69)
LQZO	6000	7.83 (8.33)	8.32 (9.11)	7.30 (7.80)	7.46 (7.91)	7.94 (8.42)	8.54 (8.94)

spectra can also occur without quantization. Together, these comparisons indicate that scale parameterization alone does not explain the observed spectra. They are empirical findings for the evaluated settings, rather than a claim that all ZO updates must have flat spectra.

Table 8: Spectral ratios for FP16 OPT-6.7B, layer 31. Entries are σ_1/σ_5 (σ_1/σ_{10}). FO denotes weight gradients and MeZO denotes cumulative weight displacements after 6000 steps.

Module	FO	MeZO
k_proj	3.23 (5.88)	1.19 (1.29)
v_proj	41.5 (75.5)	1.01 (1.02)
q_proj	4.36 (8.24)	1.16 (1.17)
out_proj	4.34 (8.51)	1.50 (1.69)
fc2	2.61 (4.31)	1.55 (1.68)

H.2 Robustness to Quantization Group Size

To assess sensitivity to scale granularity, we compare QZO and LQZO on OPT-6.7B with GPTQ INT4 and SST-2 using group sizes 32, 64, and 128. Results in Tab. 9 are means and standard deviations over three random seeds. LQZO improves accuracy for each evaluated group size, with gains of 1.30%–5.24%. This supports robustness within the tested GPTQ configurations. Our ViT experiments additionally use round-to-nearest (RTN) weight quantization, providing evidence under a second quantization method, although on a different model family and modality.

Table 9: Accuracy (%) on SST-2 with GPTQ INT4 OPT-6.7B under different group sizes. Gains (%) are the absolute differences between the reported mean accuracies. Scale shapes refer to an unreshaped 4096×4096 weight matrix. Other blocks can have different shapes.

Group size	Scale shape	QZO	LQZO	Gain
32	4096×128	89.64 ± 0.70	91.74 ± 1.23	2.10
64	4096×64	89.33 ± 0.80	90.63 ± 0.83	1.30
128	4096×32	87.65 ± 0.59	92.89 ± 0.98	5.24

H.3 Evaluation on Qwen3-4B-Base

We evaluate LQZO on Qwen3-4B-Base [53] with GPTQ INT4 and group size 128. Tab. 10 reports means and standard deviations over three random seeds, with the average taken across five tasks. LQZO improves the average score from 83.44 to 84.59, a gain of 1.15%. It outperforms QZO on RTE, WiC, SQuAD, and COPA. SST-2 scores are similar, with QZO slightly higher.

Table 10: Results on Qwen3-4B-Base with GPTQ INT4, group size 128. Classification and multiple-choice tasks use accuracy. SQuAD uses F1. Scores are on a 0–100 scale.

Method	SST-2	RTE	WiC	SQuAD	COPA	Avg.
Zero-shot (FP16)	58.83	86.28	57.21	15.91	84.00	60.45
Zero-shot (INT4)	57.68	85.92	58.15	14.85	84.00	60.12
QZO	91.67 ± 0.57	86.76 ± 0.75	66.51 ± 1.18	88.94 ± 2.37	83.33 ± 2.08	83.44 ± 1.39
LQZO	91.59 ± 0.33	88.09 ± 1.30	68.23 ± 0.33	89.02 ± 1.16	86.00 ± 3.00	84.59 ± 1.23

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the three contributions (LR-Q-SPSA, Binary-Aggregated Gaussian Perturbation, Isotropy-Enhanced Estimation), the scope of the base-estimator convergence guarantee, and the empirical results, which are supported by the theoretical analysis in Sec. 4 and experiments in Sec. 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Sec. 6 discusses convergence speed, dependence on gradient structure, the evaluated quantization configurations, untested MoE and activation-quantization settings, forward-pass memory, and the limits of theoretical hyperparameter guidance.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All assumptions are explicitly stated (Asms. 4.1 to 4.3). Proofs of asymptotic unbiasedness (Thm. G.1), a second-moment bound (Thm. G.2), and finite-bias convergence of the independently refreshed, unclipped base estimator (Thm. 4.4) are provided in App. G. Sec. G.4 explains why lazy updates and clipping are excluded from this guarantee.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: App. A provides comprehensive experimental details including model configurations, dataset splits, hyperparameter settings (Tab. 2), quantization procedures, training steps, and evaluation protocols. The algorithm is fully specified in Alg. 1. Source code is available in the repository linked in the abstract.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Source code and reproduction instructions are available at `https://github.com/WangFei-2019/UnifiedZO`. All datasets and models used are publicly available.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Sec. 5 presents the experimental setup, and App. A provides full details including data splits, hyperparameter configurations (Tab. 2), optimizer settings, quantization procedures, and evaluation metrics.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Fine-tuning task scores are averaged over three random seeds, and tables report their standard deviations ($\pm$). These standard deviations capture variability across random seeds. See Sec. 5 and App. A.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper specifies the GPU type (NVIDIA A100-40GB), reports per-step GPU timing measurements (Fig. 10), and provides peak memory usage analysis (App. E). The GPU allocation policy for large FP16 MeZO configurations is described in App. A. Total compute time across all experiments is not reported due to the large number of heterogeneous runs across different model scales and tasks.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research focuses on optimization algorithms for efficient LLM fine-tuning and conforms with the NeurIPS Code of Ethics. No human subjects, private data, or ethically sensitive applications are involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Sec. 6 includes a Broader Impacts paragraph discussing wider access to model adaptation, potential bias and misuse, and responsible downstream evaluation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper proposes an optimization algorithm and does not release pre-trained models, datasets, or other artifacts with high misuse risk.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All models (OPT, LLaMA, Qwen, ViT) and datasets (SuperGLUE benchmarks, SQuAD, DROP, CIFAR, Oxford-IIIT Pet) are properly cited with their original publications. The models and datasets used are available under their respective licenses.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The source code and reproduction documentation for LQZO are available in the repository linked in the abstract. The algorithm is fully described in Alg. 1.

Guidelines:

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

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

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

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

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

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

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

Answer: [N/A]

Justification: This paper does not involve research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs serve as evaluation subjects rather than components of the core methodology. They are not used in a non-standard way to develop the method.

Guidelines:

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