O^{2}q: Output-Space LoRA Initialization Before Quantization for Fine-Tuning Quantized LLMs

Anonymous ACL submission

Abstract

Quantized low-rank adaptation (QLoRA) enables efficient fine-tuning of large language models (LLMs), but its performance degrades sharply at lower bit-widths. Existing LoRA initialization methods for quantized LLMs differ in both what the LoRA preserves and whether LoRA initialization is performed before or after quantization, making the key design principles unclear. In this paper, we place prior SVD-based methods in a unified framework. Our analysis indicates that stronger initialization is associated with preserving the principal directions of layer outputs WX, rather than those of the weight matrix W, and with preserving before quantization so that a more concentrated residual remains to be quantized. Based on this analysis, we propose O^{2}q, an output-space LoRA initialization method for quantized LLMs that initializes the LoRA from the principal directions of WX and quantizes only the residual weight. Across Llama-3-8B and Qwen2.5-7B on mathematical reasoning and instruction-following tasks, O^{2}q achieves the best average scores among the baselines in the 2-bit setting, reaching 37.3 at rank 32 and 38.4 at rank 64.

1 Introduction

Large language models (LLMs) (Yang et al., 2024c; Grattafiori et al., 2024; DeepSeek-AI, 2026) have achieved remarkable performance across a wide range of language understanding (Zhao et al., 2023; Yang et al., 2024b) and generation tasks (Yang et al., 2024a; Guo et al., 2025), but their growing scale has also made adapting them to downstream tasks increasingly expensive. Conventional full fine-tuning updates all model parameters and requires substantial memory to store model weights and optimizer states (Liao et al., 2024), which quickly becomes impractical for models with billions of parameters. Parameter-efficient fine-tuning (PEFT) (Houlsby et al., 2019; Han et al., 2024) methods such as Low-Rank Adaptation (LoRA) (Hu et al., 2022) alleviate this burden by learning a small set of low-rank updates while keeping the pretrained backbone frozen. In a complementary direction, quantization (Frantar et al., 2022; Lin et al., 2024) reduces the memory cost of storing pretrained weights by representing them in low precision. Quantized low-rank adaptation (QLoRA) (Dettmers et al., 2023) combines these two ideas by quantizing the pretrained model while introducing trainable full-precision LoRA factors, thereby enabling memory-efficient adaptation of large language models to downstream tasks.

While QLoRA enables memory-efficient adaptation of large language models, extending this paradigm to lower bit-widths remains challenging (Li et al., 2023). In lower bit-widths, especially at 2-bit, quantization introduces a much larger discrepancy between the low-bit approximation Q and the original weight W. Standard LoRA initialization does not reduce this gap before training, leaving fine-tuning to both compensate for quantization distortion and learn task-specific adaptation (Deng et al., 2025). Consequently, finding a suitable initialization for the LoRA factors B and A becomes a central problem in lower bit-width settings.

Recent work has explored different strategies for initializing the LoRA factors in quantized LLMs. LoftQ (Li et al., 2023) addresses this issue by initializing the LoRA branch to fit the quantization residual. QPiSSA (Meng et al., 2024) extracts the dominant singular directions of the pretrained weight before quantization, leaving the remaining residual to be quantized. ApiQ (Liao et al., 2024) introduces calibration activations and a gradient-based optimization procedure to mitigate activation error propagation. CLoQ (Deng et al., 2025) also leverages calibration data, but proposes a closed-form initialization that targets layer-wise output reconstruction. However, these methods are proposed and studied as separate algorithms, leaving

it unclear which choices are most important for quantized PEFTs in lower bit-width settings.

In this work, we place existing LoRA initialization methods of quantized LLMs in a unified framework defined by two independent design axes: the allocation-target axis, which specifies what information the initialization aims to preserve, and the allocation-order axis, which determines whether LoRA initialization is performed before or after quantization. To better preserve pretrained behavior, we argue that initialization should target the layer output rather than the weight matrix itself, since layer outputs provide a better proxy for downstream task performance (Frantar et al., 2022; Frantar and Alistarh, 2023; Wang et al., 2025). Furthermore, to mitigate quantization error while enabling effective task adaptation, we propose performing LoRA initialization before quantization.

Motivated by these principles, we introduce O^2q, an output-space initialization scheme for LoRA fine-tuning of quantized LLMs. During initialization, O^2q leverages calibration activations to identify the principal directions of layer outputs, preserves these directions in the full-precision LoRA branch, and quantizes only the residual weight. We also integrate this principle into the gradient-based method, ApiQ, yielding O^2q +ApiQ. Experiments show that O^2q consistently improves downstream performance in lower bit-width settings, both as a standalone initializer and when used to initialize ApiQ.

2 Related Work

2.1 Quantization of Large Language Models

Quantization aims to reduce the memory and serving cost of large language models by mapping full-precision weights to low-bit representations. Recent post-training quantization(PTQ) methods have shown that 3- and 4-bit weight-only quantization can preserve model quality to a large extent, making low-bit deployment increasingly practical for LLM inference (Frantar et al., 2022; Shao et al., 2024; Lin et al., 2024). GPTQ (Frantar et al., 2022) applies second-order information to minimize reconstruction error. OmniQuant (Shao et al., 2024) introduces quantization parameters and optimizes them with calibration data. AWQ (Lin et al., 2024) uses calibration data to identify salient weights and protect important channels. However, pushing quantization toward lower bit-width regimes is substantially more challenging, as prior works have reported considerable performance degradation under 2-bit quantization (Frantar et al., 2022; Chee et al., 2023).

2.2 LoRA Fine-Tuning of Quantized LLMs

QLoRA (Dettmers et al., 2023) enables memory-efficient adaptation by quantizing the pretrained LLMs while training full-precision LoRA factors on top of the frozen low-bit weights. Although this paradigm is effective at 4-bit, later works (Li et al., 2023; Meng et al., 2024; Deng et al., 2025; Liao et al., 2024) find that its performance degrades substantially at lower bit-widths and that carefully initializing the LoRA branch before task-specific fine-tuning is crucial for mitigating this degradation. An important line of work uses SVD-based initialization to construct the LoRA branch before task-specific fine-tuning: LoftQ (Li et al., 2023) extracts the dominant singular directions of the quantization residual, whereas QPiSSA (Meng et al., 2024) decomposes the original weight matrix before quantization and leaves the remaining residual to be quantized. CLoQ (Deng et al., 2025) further uses calibration activations, shifting the objective from weight reconstruction to layer-wise output reconstruction. Beyond these closed-form SVD-based methods, ApiQ (Liao et al., 2024) introduces a gradient-based framework that jointly optimizes the LoRA branch and the quantized weights to reduce cross-layer activation-error propagation.

Although these approaches have demonstrated empirical gains, many of their design choices are evaluated within specific methods, making it difficult to isolate the factors that drive the improvements. A systematic comparison of the key design choices for low-bit LoRA initialization remains lacking.

3 Method

We propose O^2q, an initialization method for LoRA fine-tuning of quantized LLMs. In §3.1, we define the low-bit QLoRA initialization problem and notation. In §3.2, we introduce a unified framework that organizes prior SVD-based initialization methods and identifies the two design choices behind O^2q. In §3.3 and §3.4, we justify these choices; finally, in §3.5, we present the full algorithm and its extension to O^2q +ApiQ.

3.1 Problem Definition

Considering a pretrained linear weight $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, QLoRA (Dettmers et al., 2023) repre-

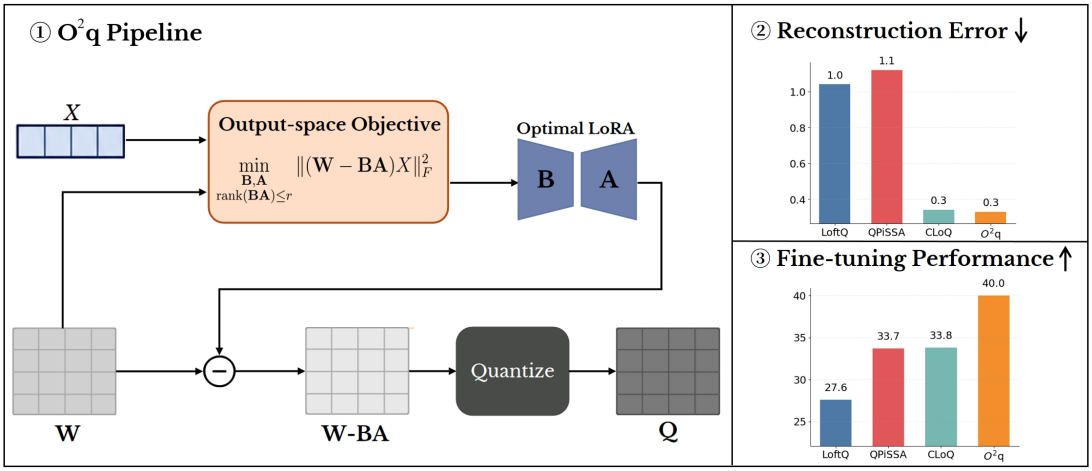

Figure 1: Overview of the O^{2}q initialization pipeline and its empirical effect. **Left**: the O^{2}q pipeline. Given weights W and calibration activations X, O^{2}q initialize the LoRA factors B and A by minimizing the output-space objective. The residual weight $W - BA$ is then quantized to produce the low-bit approximation Q. **Right**: two comparisons under the same experiment setting. The upper panel reports normalized reconstruction error after initialization. ↓ indicates that lower is better. The lower panel reports end-to-end fine-tuning performance. ↑ indicates that higher is better.

sents the adapted weight as $Q + BA$, where Q is a frozen low-bit approximation and $B \in \mathbb{R}^{d_{\text{out}} \times r}$ and $A \in \mathbb{R}^{r \times d_{\text{in}}}$ are trainable full-precision low-rank factors with $r \ll \min(d_{\text{out}}, d_{\text{in}})$. In the standard setting, Q is obtained by quantizing W and the LoRA is initialized without explicitly preserving the pretrained layer's behavior. This works well at 4-bit, but lower bit-widths leave a much larger gap between W and Q, making the choice of the initial Q, B, and A critical before downstream fine-tuning begins (Li et al., 2023; Deng et al., 2025; Liao et al., 2024). Our goal is therefore to choose initial Q, B, and A that retain as much of the pretrained layer's behavior as possible while leaving the LoRA well positioned for subsequent task adaptation.

3.2 Unified framework

We propose a unified framework that places prior SVD-based LoRA initialization methods under the same low-rank reconstruction objective. Let T denote the matrix that the LoRA is initialized to approximate, and let $S \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ denote the calibration transform that specifies how reconstruction error is measured. Within this framework, SVD-based initialization reduces to the following calibrated reconstruction problem:

$$\min_{B,\, A:\ \text{rank}(BA) \le r} \left\| (T - BA)\, S \right\|_F^2. \tag{1}$$

Existing closed-form SVD initialization methods differ by two independent design choices.

Axis 1: Pipeline order. The first choice determines the LoRA target T. In *quantize-then-BA* methods (Li et al., 2023; Deng et al., 2025), the pretrained weight is first quantized and the LoRA is initialized to fit the quantization residual:

$$Q = \text{Quantize}(W), \qquad T = W - Q. \tag{2}$$

In *BA-then-quantize* methods (Meng et al., 2024), the LoRA factors are initialized before quantization and directly approximate the pretrained weight:

$$T = W. \tag{3}$$

The residual $W - BA$ is then quantized as Q.

Axis 2: Distance metric. The second choice determines which error the LoRA initialization preserves. *Weight-space* methods set $S = I$, yielding

$$\mathcal{L}_{\text{w}}(T;\, B, A) = \|T - BA\|_F^2, \tag{4}$$

which treats all weight directions as equally important. *Output-space* methods use calibration activations $X \in \mathbb{R}^{d_{\text{in}} \times n}$ and set $SS^\top = XX^\top$, yielding

$$\mathcal{L}_{\text{x}}(T;\, B, A) = \|(T - BA)X\|_F^2, \tag{5}$$

so that errors are weighted by the input distribution.

In this view, the target T encodes *when* the low-rank component is allocated relative to quantization, while the calibration transform S specifies *what notion of importance* is used when selecting the rank-r subspace. Table 1 places prior methods

Method	Order	Target T	Metric	Objective
LoftQ	Q→BA	$W - \text{Quantize}(W)$	weight	$\|T - BA\|_F^2$
QPiSSA	BA→Q	W	weight	$\|T - BA\|_F^2$
CLoQ	Q→BA	$W - \text{Quantize}(W)$	output	$\|(T - BA)X\|_F^2$
O^2q	BA→Q	W	output	$\|(T - BA)X\|_F^2$

Table 1: Unified framework for closed-form SVD-based LoRA initialization. The Order column uses Q→BA for quantize-then-BA and BA→Q for BA-then-quantize. ApiQ also targets output space but optimizes via gradient descent rather than the closed form in Eq. (6); see §3.5.

and O^2q in a unified design space defined by the choice of target T and distance metric, highlighting the distinct design principles underlying each method.

The unified objective Eq. 1 further yields a shared closed-form solution for initializing the LoRA factors. For any target T specified by the pipeline order in (2) or (3), (1) is equivalent to approximating TS with a rank-r matrix. By the Eckart–Young–Mirsky theorem (Eckart and Young, 1936), the optimal approximation is given by the rank-r truncated SVD of TS. Writing $[U, \Sigma, V^\top] = \text{SVD}(TS)$, and mapping the approximation back through $S^\dagger$, one optimal choice of factors is

$$B = U_{[:,:r]}\, \Sigma_{[:r]}^{1-\alpha}, \qquad A = \Sigma_{[:r]}^{\alpha}\, V_{[:r,:]}^\top\, S^\dagger, \quad (6)$$

where $\alpha \in [0, 1]$ distributes the singular values between B and A, and $S^\dagger$ denotes the Moore–Penrose pseudoinverse of S.

3.3 Output-Space Allocation

Ideally, initialization should preserve the behavior of the pretrained layer, not merely the entries of its weight matrix. Given activations X, the layer output before initialization is WX, while the initialized quantized-LoRA layer outputs $(Q+BA)X$. The natural objective is therefore to reduce the layer-wise output reconstruction error $\|(W - Q - BA)X\|_F^2$. In the unified framework above, the quantity to be allocated to the LoRA is denoted by T; equivalently, we minimize

$$\mathcal{L}_{\text{out}}(T;\, B, A) = \|(T - BA)X\|_F^2, \quad (7)$$

which directly measures the discrepancy seen by subsequent layers and thus serves as a closer proxy for model behavior than weight-space error alone. This choice is also consistent with the broader practice in model compression (Frantar et al., 2022; Frantar and Alistarh, 2023; Lin et al., 2024; Wang et al., 2025).

Weight-space allocation is suboptimal for this objective because it corresponds to setting $S = I$ in Eq. (1). In that case, the initialization minimizes $\|T - BA\|_F^2$ and allocates rank according to the singular directions of the target matrix itself. This is equivalent to minimizing output reconstruction only under the isotropic assumption $XX^\top \propto I$, where all input directions have equal covariance and output-space error becomes a constant multiple of weight-space error. However, in practice, LLM activations are highly anisotropic (Godey et al., 2024; Rudman and Eickhoff, 2024), as we verify empirically in Section 4.4.1. Therefore, different input directions contribute very unequally to the layer output, while weight-space allocation implicitly treats them as equally important. Output-space allocation addresses this mismatch by choosing $SS^\top = XX^\top$, so the rank budget is assigned according to the directions that actually dominate WX.

3.4 Low-Rank Initialization Before Quantization

§3.2 established *what* the LoRA should preserve, the principal directions of WX; we now argue for *when* this allocation should occur, specifically, that the low-rank branch should be initialized before quantization.

The residual is easier to quantize. In quantize-then-BA methods, B and A are initialized to approximate the principal directions of $(W - Q)X$, after the full W has already been exposed to quantization. Under the BA-then-quantize order, the principal directions of WX are preserved to the full-precision branch BA before quantization, leaving a residual $W - BA$ with a more concentrated weight distribution. We measure this effect using the group-wise dynamic range (GDR), defined as the mean per-group $(\max - \min)$ over all quantization groups; lower GDR indicates a tighter weight distribution and lower quantization error at the same bit-width. We evaluate GDR together with the normalized output reconstruction error $\frac{\|(W-Q-BA)X\|_F^2}{\|WX\|_F^2}$ across all methods in §4.4.2.

Output-aligned initialization. Low-rank initialization before quantization can also place the LoRA factors in a better position for fine-tuning. During fine-tuning, Q is frozen and only B and A are up-

Algorithm 1 Output-Space Low-Rank Initialization Before Quantization (O^2q)

Require: Weight W, calibration activations X, rank r, iterations K
Ensure: Quantized weight Q, LoRA factors B, A

$H \leftarrow XX^\top$
$H \leftarrow H + \lambda I,\ \lambda = \frac{0.01\,\mathrm{Tr}(H)}{m}$
$[U_H,\, \Sigma_H] \leftarrow \mathrm{Eig}(H);\ S \leftarrow U_H \Sigma_H^{1/2}$
$S^\dagger \leftarrow \Sigma_H^{-1/2} U_H^\top$
$Q \leftarrow \mathbf{0}$ ▷ BA-then-quantize order
for $k = 1$ **to** K **do**
  $T \leftarrow W - Q$
  $[U,\, \Sigma,\, V^\top] \leftarrow \mathrm{SVD}(TS)$
  $B \leftarrow U_{[:,\,:r]};\ A \leftarrow \Sigma_{[:r]}\, V^\top_{[:r,\,:]}\, S^\dagger$ ▷ LoRA step
  $Q \leftarrow \mathrm{Quantize}(W - BA)$ ▷ quantization step
end for
return $Q,\ B,\ A$

dated. The gradients of B and A are

$$\frac{\partial \mathcal{L}}{\partial B} = \frac{\partial \mathcal{L}}{\partial Y} X^\top A^\top, \qquad \frac{\partial \mathcal{L}}{\partial A} = B^\top \frac{\partial \mathcal{L}}{\partial Y} X^\top$$

so A's row space determines which input directions are projected onto B's update, and B's column space determines which output directions are projected onto A's update. When the initialization is output-space scheme and follows a BA-then-quantize order, A is set via (6) so that its rows span the dominant input directions of WX, and B is initialized to the corresponding left singular vectors; both projections are therefore concentrated on the directions that carry high output energy (§3.3). In quantize-then-BA methods, both B and A are initialized to approximate the dominant directions of quantization residual, $(W - Q)X$. Early optimization may therefore spend part of its capacity reorienting the LoRA subspace before adapting to the downstream objective. This is consistent with PiSSA's observation that principal-subspace initialization reduces the reorientation cost in early training (Meng et al., 2024); empirically, we find that BA-then-quantize methods reach lower training loss faster than quantize-then-BA methods in §D.

3.5 O^2q Algorithm

Algorithm 1 instantiates the two design choices above: it initializes the LoRA factors from the principal directions of WX and quantizes only the residual $W - BA$. The calibration matrix S is constructed from the eigendecomposition $H = U_H \Sigma_H U_H^\top$ as $S = U_H \Sigma_H^{1/2}$ and, in the regularized setting below, $S^\dagger = \Sigma_H^{-1/2} U_H^\top$. Before factorization, we regularize H as $H \leftarrow H + \lambda I$ with $\lambda = 0.01\,\frac{\mathrm{Tr}(H)}{m}$, where m is the dimension of H. This standard stabilization step improves conditioning and makes the resulting S nonsingular in practice (Frantar et al., 2022; Chee et al., 2023).

Extension to O^2q +ApiQ. ApiQ (Liao et al., 2024) optimizes $\min_{Q,\,B,\,A} \|WX - (Q + BA)\hat{X}\|_F^2$ in a transformer block to address cross-layer quantization error propagation, where $\hat{X}$ denotes the perturbed activations propagated through preceding quantized layers. It jointly refines Q and BA via gradient descent with a straight-through estimator (Bengio et al., 2013). Its LoRA factors are initialized with $B = \mathbf{0}$, $A \sim \mathcal{N}(0, 1)$.

O^2q +ApiQ replaces this standard initialization with the LoRA step in Algorithm 1, providing a stronger starting point for gradient refinement. The two methods address complementary limitations: O^2q improves the per-layer starting point by aligning the LoRA factors with the dominant directions of WX before quantization, whereas ApiQ further corrects cross-layer error propagation by jointly refining Q and BA across transformer blocks.

4 Experiments

4.1 Experimental Setup

Models and tasks. We evaluate on **Llama-3-8B** (Grattafiori et al., 2024) and **Qwen2.5-7B** (Yang et al., 2024c) over two downstream domains. For *mathematical reasoning*, models are fine-tuned on MetaMathQA (Yu et al., 2024) and

		Llama-3-8B					Qwen2.5-7B					
		Math		Instruction			Math		Instruction			
Rank	Method	GSM8K	MATH	IFEval	HSR	SSR	GSM8K	MATH	IFEval	HSR	SSR	Avg.
	Base	37.2	11.7	25.3	15.2	24.0	76.6	40.0	31.8	41.3	54.0	35.7
32	LoftQ	36.9	7.6	21.4	13.1	27.8	61.5	23.5	27.9	22.1	36.9	28.9
	QPiSSA	49.1	11.2	23.5	17.5	34.7	67.0	28.4	28.5	26.2	42.5	35.3
	CLoQ	49.4	12.1	21.6	16.9	31.7	71.7	33.3	29.6	23.4	40.5	33.0
	ApiQ	56.0	14.6	28.3	21.2	37.3	75.4	35.5	29.6	**27.6**	43.5	36.9
	O^2q	58.1	14.9	29.9	19.1	37.3	74.8	35.7	31.8	27.7	43.4	37.3
	O^2q +ApiQ	**60.3**	**15.5**	**30.7**	**21.9**	**39.3**	**75.9**	**36.9**	**30.1**	25.8	**43.6**	**38.0**
64	LoftQ	45.5	9.6	24.4	15.4	32.2	65.2	26.2	28.1	22.4	39.2	30.8
	QPiSSA	55.2	12.2	27.4	18.6	36.6	72.3	30.5	29.6	24.4	41.7	34.9
	CLoQ	53.9	13.7	24.8	21.1	37.0	73.5	34.0	27.5	27.0	42.4	35.5
	ApiQ	60.6	15.9	29.6	22.7	38.3	**76.4**	36.5	26.8	29.3	**46.8**	38.3
	O^2q	63.5	16.6	28.8	20.2	38.0	74.6	35.3	**31.8**	**29.6**	46.0	38.4
	O^2q +ApiQ	**64.1**	**17.8**	**32.0**	**25.2**	**41.9**	76.1	**36.7**	30.3	29.1	45.5	**39.9**

Table 2: The 2-bit main results across math and instruction following. Best quantized result is reported in **bold**.

evaluated on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). For *instruction following*, models are fine-tuned on Conifer (Sun et al., 2024)+ShareGPT[1] (Chiang et al., 2023) and evaluated on IFEval (Zhou et al., 2023) and FollowBench (Jiang et al., 2024).

Baselines. We compare against the standard quantized LoRA initialization (**QLoRA** (Dettmers et al., 2023)), the weight-space SVD methods **LoftQ** (Li et al., 2023) and **QPiSSA** (Meng et al., 2024), the output-space method **CLoQ** (Deng et al., 2025), and the gradient-based method **ApiQ** (Liao et al., 2024). We report **O^2q +ApiQ**, which replaces ApiQ's default LoRA initialization with the SVD step of O^2q (§3.5). The original model performance (Base) is also reported in the experiments. Complete implementation and evaluation details, including the metrics and the different hyperparameters used by ApiQ and the SVD-based initializers, are provided in Appendix A.

Fine-tuning and evaluation. For all methods and downstream domains, we fine-tune with Adam using learning rate 5×10^{-5}, a cosine schedule with warmup ratio 0.03, batch size 128. We train for one epoch over 100K data for mathematical reasoning and 107K data for instruction following. We use sequence length 512 for mathematical reasoning and 2048 for instruction following. For instruction following, we apply each model's native chat template.

[1] https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered/blob/main/ShareGPT_V3_unfiltered_cleaned_split.json

4.2 Main Results

Table 2 reports that O^2q achieves the best average score among all the baselines, reaching 37.3 at rank 32 and 38.4 at rank 64. O^2q +ApiQ further gives the strongest overall result, raising the average to 38.0 and 39.9 at the two ranks. These results establish the effectiveness of O^2q in the low-bit regime and suggest that the same initialization principle can also extend to the gradient-based method.

Compared with CLoQ, O^2q highlights the benefit of BA-then-quantize order under the same output-space objective. O^2q improves the average from 33.0 to 37.3 at rank 32 and from 35.5 to 38.4 at rank 64. Furthermore, O^2q outperforms CLoQ on all $10/10$ task metrics at rank 32 and on $9/10$ metrics at rank 64. This shows that preserving the LoRA before quantizing is a more effective way.

Compared with QPiSSA, O^2q improves the average from 35.3 to 37.3 at rank 32 and from 34.9 to 38.4 at rank 64, and it is better on all $10/10$ task metrics at both ranks. These results highlight the benefit of LoRA initialization in output-space and the output-space allocation yields consistent task-level gains across both models and both domains in the 2-bit setting.

4.3 Ablations

Components ablation. To assess the contribution of each component in O^2q to the final performance gain, we perform a cumulative ablation that starts from the full method O^2q, we sequentially replace or remove components and observe the resulting drop. As shown in Table 3, each change leads to a further decrease. Replacing the output-

Variant	Llama-3-8B		Qwen2.5-7B		Avg.
	GSM8K	MATH	GSM8K	MATH	
O^2q	**63.5**	**16.6**	**74.6**	**35.3**	**47.5**
↪ NO output-space	55.2	12.2	72.3	30.5	42.6
↪ NO init-before-quant	45.5	9.6	65.2	26.2	36.6
↪ NO SVD init	37.7	7.9	58.2	21.7	31.4

Table 3: Component ablation on 2-bit mathematical reasoning at rank 64. Each row cumulatively removes one design choice from the row above.

Selection	Llama-3-8B		Qwen2.5-7B		Avg.
	GSM8K	MATH	GSM8K	MATH	
Top-r	**63.5**	**16.6**	**74.6**	**35.3**	**47.5**
Middle-r	52.4	12.3	71.9	33.0	42.4
Tail-r	52.3	12.9	73.3	31.8	42.6
Random-r	50.6	13.5	72.3	33.4	42.5

Table 4: Different rank selection strategies ablation on 2-bit mathematical reasoning at rank 64. Top-r selects the first r directions in descending singular-value order. Middle-r denotes the contiguous rank-r slice starting one-third of the way, Tail-r denotes the last rank-r slice, and Random Selection uniformly samples a set of r directions from the singular space.

space allocation reduces the average score from 47.5 to 42.6. Next, altering the order of initialization and quantization further lowers the score to 36.6. Finally, reverting to standard LoRA initialization drops the score to 31.4. The monotonic degradation across these four configurations confirms that the full initialization design of O^2q is critical in the 2-bit setting.

Singular-direction selection ablation. To verify whether the gain comes specifically from preserving the dominant output-space singular directions, rather than from using another rank-64 selection of the same size, we evaluate different selection strategies under the same 2-bit mathematical reasoning setting on Llama-3-8B and Qwen2.5-7B. Here, Top-r denotes the O^2q selection, which selects the first r directions in descending singular-value order. Middle-r denotes the contiguous rank-r slice starting one-third of the way, Tail-r denotes the last rank-r slice, and Random Selection uniformly samples a set of r directions from the singular space. As shown in Table 4, Top-r gives the best average performance at 47.5, while the other three selections cluster at 42.4–42.6 average score. These results show that the gain depends on preserving the dominant output directions rather than on using the same rank size.

Experiments using more data. We study whether the initialization advantage persists under

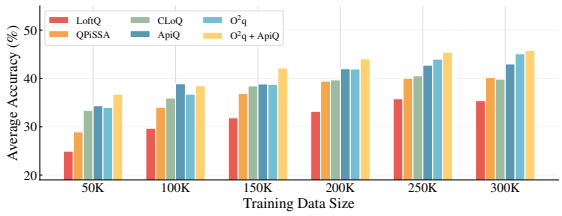

Figure 2: The 2-bit fine-tuning results on mathematical reasoning using Llama-3-8B with up to 300K training examples. We evaluate the performance every 50k training samples and report the average score on GSM8K and MATH.

Model	Math	Instruct
Llama-3-8B	0.028	0.071
Qwen2.5-7B	0.024	0.048

Table 5: IsoScore $\iota(X)$ of calibration activations, averaged over the transformer blocks.

a longer training budget. Fig 2 reports the fine-tuning data size from 50k to 300k samples on 2-bit Llama-3-8B math every 50k samples. At the first 50k checkpoint, O^2q already starts above its baseline CLoQ, while O^2q +ApiQ also starts above ApiQ, showing that the benefit of a stronger initialization appears from the early stage of fine-tuning. Through the middle of training, this advantage does not disappear: both O^2q and O^2q +ApiQ remain above their baselines for most of the trajectory. By the final 300k checkpoint, O^2q still remains above CLoQ and O^2q +ApiQ achieves the best performance overall. These results show that the advantage of better initialization is not confined to the short training budget used in the main table, but continues to translate into stronger performance under a longer training setting. We also report the GSM8K and Math acc score in the appendix C.

4.4 Analysis

4.4.1 Anisotropy of Calibration Activations

In this section, we analyze the isotropy of calibration activations using IsoScore (Godey et al., 2024), denoted by the anisotropy metric $\iota(X)$, where ι=1 indicates perfect isotropy and ι→0 indicates strong concentration in a small number of directions. Table 5 reports the mean IsoScore across transformer blocks for Llama-3-8B and Qwen2.5-7B on the mathematical reasoning and instruction following settings. The values range from 0.024 to 0.071, all far below 1, showing that the calibration activations are concentrated on a small number of dominant directions rather than being isotropic.

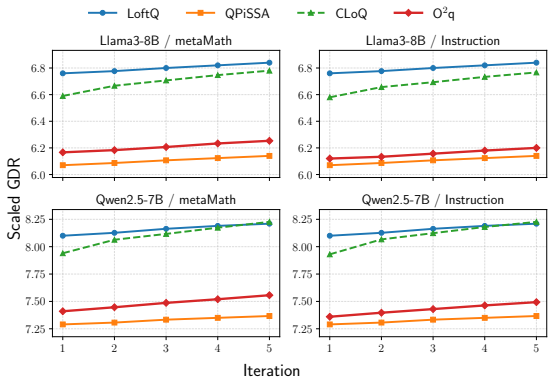

Figure 3: Group-wise dynamic range (GDR) of the quantization target during initialization. The y-axis reports $\mathrm{GDR} \times 100$.

4.4.2 Dynamic Range and Reconstruction Error

Group-wise dynamic range. We analyze the distribution of quantization target by the group-wise dynamic range (GDR), which is defined as the mean per-group $(\max - \min)$ over the groups, and the lower means a tighter per-group value range. As shown in Figure 3, BA-then-quantize methods achieve lower GDR than quantize-then-BA methods throughout all iterations across both models and domains, demonstrating that initializing LoRA first leaves a more concentrated residual to quantize.

Normalized output reconstruction error. We utilize the normalized output reconstruction error, $\frac{\|(W-Q-BA)X\|_F^2}{\|WX\|_F^2}$, to measure the initialization quality of each methods. As shown in Figure 4, O^{2}q achieves the lowest error among the methods. On average over the two models, O^{2}q reduces the reconstruction error from 0.894 to 0.284 compared with QPiSSA, and from 0.295 to 0.284 compared with CLoQ. These results indicate that combining output-space allocation with BA-quantize order can better preserve the task-relevant output information in WX.

To further connect this error with model behavior, we evaluate the quantization quality and downstream fine-tuning performance of Llama-3-8B across $\{2, 3, 4\}$-bit mathematical reasoning settings. The results show that O^{2}q and O^{2}q +ApiQ are most beneficial in the low-bit regime. Details are reported in Appendix B.

4.4.3 Calibration sample size.

As O^{2}q estimates the Gram matrix $H = XX^\top$ from calibration activations, we evaluate its sensitivity to calibration set size. In the 2-bit mathematical reasoning setting, we vary the number of calibration sequences from 32 to 256 while keeping all other hyperparameters fixed. Figure 5 shows that performance improves up to 64 calibration samples and then largely plateaus, indicating that a small calibration set is sufficient to recover most of the initialization benefit.

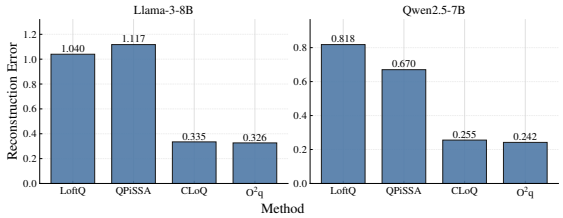

Figure 4: Normalized output reconstruction error after initialization at 2-bit on MetaMath calibration activations. Lower is better.

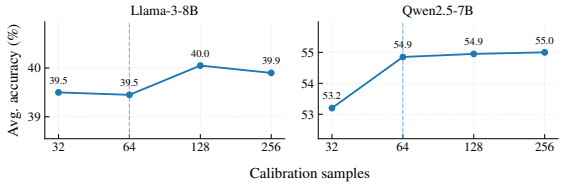

Figure 5: Effect of calibration sample size on 2-bit math performance. Each panel reports the average of GSM8K and MATH for one base model as the number of calibration samples varies.

5 Conclusion

In this study, we present O^{2}q, an output-space initialization before quantization for the quantized LLMs method. Within this unified framework, we analyze the difference between weight-space and output-space allocation, and suggest that the output-space objective better preserves layer-output information. We further find that initializing LoRA before quantization can reduce quantization error and provide a better starting point for fine-tuning. Experiments on Llama-3-8B and Qwen2.5-7B across mathematical reasoning and instruction following tasks indicate that, in the 2-bit setting, O^{2}q achieves improvements over existing SVD-based LoRA initialization methods for quantized LLMs, such as QPiSSA and CLoQ. Further experiments demonstrate that the design principles of O^{2}q can be extended to the gradient-based method ApiQ, yielding O^{2}q +ApiQ, which further boosts the performance of ApiQ. We believe that O^{2}q provides an effective LoRA initialization method for quantized LLMs, especially in low-bit quantization regimes.

Limitations

Due to limited computational resources, we primarily evaluate O^2q and other baselines on Qwen2.5-7B and Llama-3-8B, focusing on math and instruction following tasks. Moreover, while our empirical analysis indicates that the output-space initialization better preserves information from downstream activations, the exact relationship between this preserved information, downstream task performance, and the final fine-tuning results remains unclear. An open question for future work is to provide a more rigorous theoretical understanding of why output-space initialization translates to improved performance in the weight-space.

References

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. 2023. Quip: 2-bit quantization of large language models with guarantees. *Advances in neural information processing systems*, 36:4396–4429.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Yanxia Deng, Aozhong Zhang, Selcuk Gurses, Naigang Wang, Zi Yang, and Penghang Yin. 2025. Cloq: Enhancing fine-tuning of quantized llms via calibrated lora initialization. *arXiv preprint arXiv:2501.18475*.

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

Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

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

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Nathan Godey, Éric Clergerie, and Benoît Sagot. 2024. Anisotropy is inherent to self-attention in transformers. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 35–48.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

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

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

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688, Bangkok, Thailand. Association for Computational Linguistics.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. 2023. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*.

Baohao Liao, Christian Herold, Shahram Khadivi, and Christof Monz. 2024. Apiq: Finetuning of 2-bit quantized large language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20996–21020.

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

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072.

William Rudman and Carsten Eickhoff. 2024. Stable anisotropic regularization. In *International Conference on Learning Representations*, volume 2024, pages 23594–23609.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Gao Peng, Yu Qiao, and Ping Luo. 2024. Omniquant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, volume 2024, pages 45472–45496.

Haoran Sun, Lixin Liu, Junjie Li, Fengyu Wang, Baohua Dong, Ran Lin, and Ruohui Huang. 2024. Conifer: Improving complex constrained instruction-following ability of large language models. *Preprint*, arXiv:2404.02823.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2025. Svd-llm: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations*, volume 2025, pages 19299–19319.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. 2024a. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024b. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. 2024c. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, volume 2024, pages 45040–45061.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2):1–124.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *Preprint*, arXiv:2311.07911.

A Experimental Details

For math tasks, we report accuracy. For instruction-following evaluation, we use prompt-level loose accuracy on IFEval, and the L1–L5 average Hard Satisfaction Rate (HSR) and Soft Satisfaction Rate (SSR) on FollowBench (Jiang et al., 2024).

We follow the baseline settings as closely as possible and only change the initialization procedure being evaluated. For all methods, LoRA modules are attached to the $\{q, k, v, o, \text{gate}, \text{up}\}$ projections. Unless otherwise stated, the LoRA rank is $r{=}64$ and the LoRA scaling is 64.

For closed-form SVD-based initializers, including LoftQ, QPiSSA, CLoQ, and O^{2}q, we alternate between updating Q and BA for $T{=}5$ initialization iterations. For the exponent α in Eq. (6), LoftQ and QPiSSA use $\alpha{=}0.5$, whereas CLoQ and O^{2}q use $\alpha{=}1$, following their original settings. CLoQ and O^{2}q are calibration-based and use GPTQ; QLoRA, LoftQ, and QPiSSA use uniform quantization. For all quantization methods, we use per-channel quantization, activation-order reordering, and group size 64. For the output-space methods, we sample 128 training sequences of length 2048 with seed 0 as calibration data.

ApiQ uses a different initialization and optimization recipe from the closed-form SVD-based methods. We follow the original ApiQ setting: the LoRA factors are initialized with $B = \mathbf{0}$ and $A \sim \mathcal{N}(0, 1)$, and are then jointly refined with the quantized weights through gradient-based optimization using a straight-through estimator. In our implementation, the LoRA parameters are optimized with learning rate 1.25×10^{-4} and weight decay 0.1, while the learnable weight-clipping parameters are optimized with learning rate 5×10^{-3} and weight decay 0.1. O^{2}q +ApiQ keeps ApiQ's quantizer and gradient-based refinement procedure unchanged, but replaces ApiQ's default LoRA initialization with the O^{2}q initialization before running the same ApiQ refinement stage.

B Quantization Quality and Training Across Bit-widths

We further study how quantization quality and downstream training behavior change across bit-widths. Table 6 evaluates the math performance of the quantized models immediately after initialization, without applying the full downstream fine-tuning recipe used in the main results. This setting isolates the quality of the quantized initialization itself. The results show that the advantage of O^{2}q and O^{2}q +ApiQ is most pronounced in the challenging 2-bit setting, especially on Qwen2.5-7B, while the gap among different methods becomes smaller as the bit-width increases to 3 and 4 bits. Figure 6 complements this initialization-only diagnostic by reporting the fine-tuned math performance of Llama-3-8B at rank 64 across the same bit-widths. The trend is consistent with the reconstruction and initialization results: low-bit settings benefit more from the proposed initialization, whereas higher bit-widths reduce the performance differences among methods.

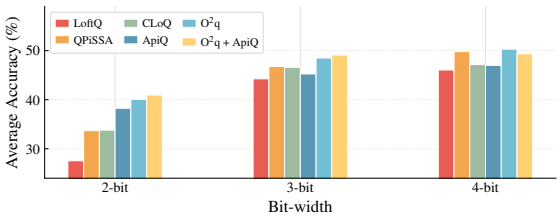

Figure 6: Effect of bit-width on math performance of Llama-3-8B at rank 64.

		Llama-3-8B		
Bit	Method	GSM8K	MATH	Avg.
2	LoftQ	0.9	1.2	1.1
	QPiSSA	0.0	0.2	0.1
	CLoQ	2.3	2.3	2.3
	ApiQ	13.6	**4.2**	8.9
	O^{2}q	0.9	1.1	1.0
	O^{2}q +ApiQ	**14.4**	3.8	**9.1**
3	LoftQ	5.4	3.3	4.4
	QPiSSA	10.1	4.1	7.1
	CLoQ	31.8	9.0	20.4
	ApiQ	29.0	8.0	18.5
	O^{2}q	**34.8**	**9.4**	**22.1**
	O^{2}q +ApiQ	33.1	9.0	21.1
4	LoftQ	30.6	9.8	20.2
	QPiSSA	32.7	10.3	21.5
	CLoQ	**35.9**	10.5	**23.2**
	ApiQ	31.1	9.7	20.4
	O^{2}q	35.1	**10.9**	23.0
	O^{2}q +ApiQ	34.7	9.8	22.3

Table 6: Math performance on Llama-3-8B immediately after quantized initialization. The table evaluates initialized quantized models without the full downstream fine-tuning recipe used in the main results. Avg. is the mean of GSM8K and MATH.

C Fine-tuning Data-size Progression

We further evaluate 2-bit Llama-3-8B on math tasks with training sets ranging from 50K to 300K examples. This experiment examines whether the advantage of O^{2}q persists when the fine-tuning budget is increased beyond the default 100K setting. Figures 7 and 8 report the results on GSM8K

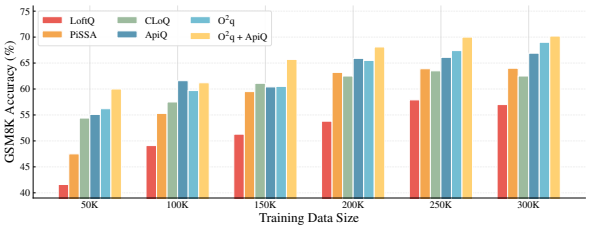

Figure 7: **GSM8K accuracy across fine-tuning data sizes for 2-bit Llama-3-8B.** Each group corresponds to a checkpoint from 50k to 300k training samples. O^2q and O^2q +ApiQ continue improving with larger training budgets, whereas LoftQ saturates earlier.

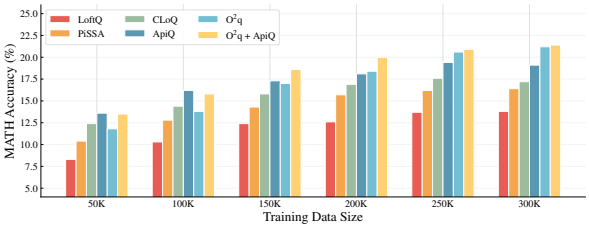

Figure 8: **MATH accuracy across fine-tuning data sizes for 2-bit Llama-3-8B.** The harder MATH benchmark shows the same broad pattern as GSM8K: ApiQ is strong at small budgets, while O^2q catches up at larger budgets and O^2q +ApiQ gives the strongest overall trajectory.

and MATH, respectively.

D Fine-tuning Loss

We examine training loss during 2-bit, rank-64 fine-tuning of Llama-3-8B on MetaMathQA under the 300K-example training setting. Figure 9 shows the loss curves over the first 100K training examples, allowing us to compare the early optimization behavior across methods. At the first logged step, the losses are 6.33 for LoftQ, 5.66 for QPiSSA, 3.06 for CLoQ, 1.37 for O^2q, 0.78 for ApiQ, and 0.74 for O^2q +ApiQ. Thus, O^2q starts from a much lower loss than the closed-form SVD baselines QPiSSA, LoftQ, and CLoQ, while O^2q +ApiQ further reduces ApiQ's first-step loss. The subsequent loss curves show the same tendency, suggesting that the initialization advantage persists during early fine-tuning, rather than only at the first optimization step.

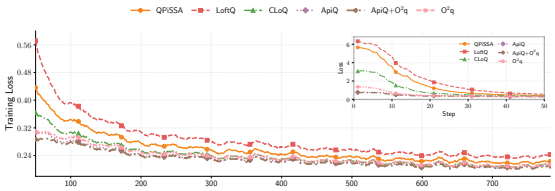

Figure 9: Fine-tuning loss on MetaMathQA for 2-bit Llama-3-8B. All methods use rank 64 and are trained with the 300K-example MetaMathQA setting. The plot reports the loss curves over the first 100K training examples. The main panel shows the loss curves after the first 50 training steps, while the inset shows the first 50 steps omitted from the main panel.