ROQAT: A ROBUST QUANTIZATION-AWARE TRAINING METHOD FOR LLMS EXTREMELY LOW-BIT DOWNSTREAM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantization-aware training (QAT) can largely recover the capabilities of low-bit large language models (LLMs). However, we discover that although achieving nearly lossless performance on general benchmarks, there still exists a tremendous gap on downstream reasoning capabilities at extremely low-bit, especially at 2-bit. We argue that this degradation is closely related to two challenges in extremely low-bit QAT: 1) directly perform low-bit training will cause severe initial degradation. Although progressive quantization can partially stabilize training, existing methods typically ignore sensitivity differences during progression; 2) the weights may converge to the sharp regions of the loss landscape, which will limit the robustness and generalization of the quantized model. Motivated by these observations, we propose **RoQAT**, a robust QAT framework for preserving downstream capabilities of extremely low-bit LLMs. RoQAT first introduces *sensitivity-adaptive progressive quantization* (SAPQ), which considers distinct quantization sensitivity across groups. Specifically, RoQAT dynamically assigns each group a different progression rate to the target bit-width according to the block quantization sensitivity along the actual progressive training direction, which allows more sensitive groups to approach the target bit-width more conservatively. Moreover, RoQAT incorporates *quantization-aware sharpness minimization* (QSAM) to establish flatter loss landscapes around low-precision weights, which further enhances downstream generalization. It performs sharpness-aware optimization in the weight space during block-wise training and the scale space during end-to-end refinement. Extensive experiments on Qwen3 instruction-tuned and Qwen3.5 hybrid-attention LLMs demonstrate that RoQAT consistently outperforms strong QAT baselines on downstream benchmarks while remaining stable general performance at extremely low-bit. Codes will be released soon.

1 INTRODUCTION

Large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023; Yang et al., 2025) have demonstrated remarkable capabilities across diverse tasks, including language understanding (Radford et al., 2018) and complex reasoning (Fu et al., 2022). However, their large parameter scales impose insufferable memory and computational overheads during deployment. To mitigate these bottlenecks, quantization (Lang et al., 2024; Li et al., 2021; Nagel et al., 2020) has recently attracted significant attention as a principal solution. Existing approaches can be broadly categorized into post-training quantization (PTQ) (Frantar et al., 2022; Lin et al., 2024; Zandieh et al., 2026) and quantization-aware training (QAT) (Liu et al., 2024b; Ma et al., 2024). Although PTQ achieves rapid compression with minimal compute overhead, its constrained optimization landscape hinders performance recovery at lower-bit. In contrast, QAT achieves much better performance by optimizing LLMs with low-bit constraints, so it exhibits superior potential for extremely low-bit quantization.

Although recent QAT methods have largely recovered the performance of extremely low-bit LLMs on general benchmarks (Du et al., 2024; Chen et al., 2025), we observe that they still suffer severe degradation in complex downstream reasoning capabilities (Li et al., 2025; Liu et al., 2025), such as mathematical reasoning (Cobbe et al., 2021) and code generation (Chen et al., 2021; Wang & Chen, 2023), as shown in Figure 1. ReasoningQAT (Okoshi et al., 2026) has taken an important step

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

to this problem by proposing a specific training dataset and reinforcement-like objective function, but a large performance gap still remains under extremely low-bit scenario. Since these capabilities are often strengthened through expensive post-training, recovering them again after quantization through additional SFT (Li & Liang, 2021; Hu et al., 2021) or RL (Shao et al., 2024; Yu et al., 2026) will introduce considerable training cost. This motivates us to ask: **can QAT itself better preserve post-trained downstream capabilities when training at extremely low-bit?**

Addressing this challenge requires rethinking standard QAT through two core dimensions: training trajectory stabilization and weight robustness enhancement. *First*, directly projecting the model into extremely low-bit at the initial phase of training will introduce significant performance collapse, which makes the subsequent QAT process difficult to optimize stably. A natural way to stabilize this optimization is progressive quantization (Tu et al., 2025; Wang et al., 2026; Xu et al., 2026), which gradually increases the quantization difficulty during training. However, existing progressive schemes typically rely on predefined precision schedules or quantizer annealing following a uniform progression process. Our analysis reveals pronounced sensitivity heterogeneity across quantization groups, as shown in Figure 3, where the same progression step may introduce different reconstruction error. This suggests adapting the progression rate of each group according to its quantization sensitivity. *In addition*, extremely low-bit QAT can converge to sharp regions of the loss landscape, as indicated by Figure 4, where small perturbations can cause severe loss spikes. Such sharpness not only makes low-precision weights less robust, but is also closely associated with worse impact on generalization (Keskar et al., 2016), which undermines the preservation of downstream capabilities. This motivates incorporating sharpness-aware optimization into QAT to guide the training to flatter regions with improved robustness and generalization.

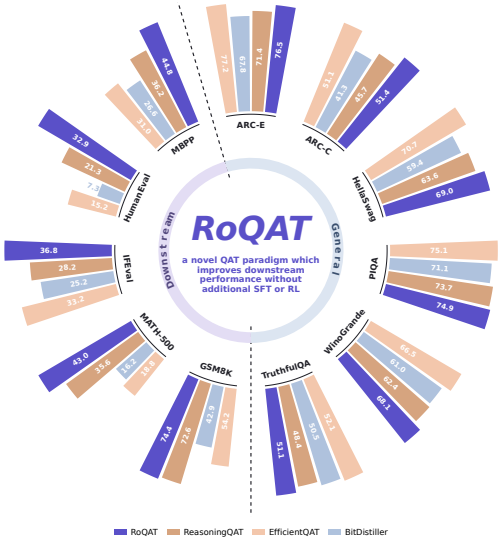
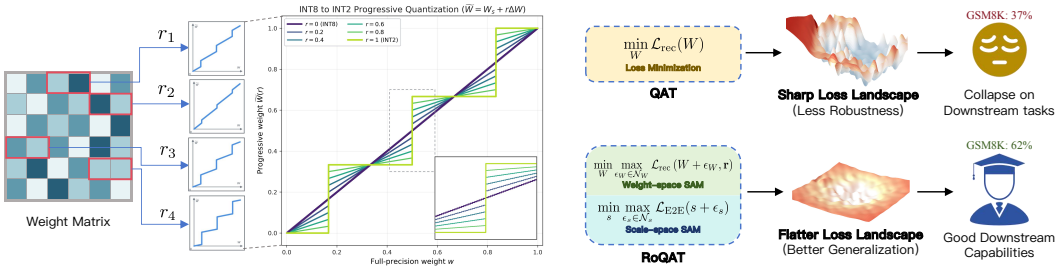


Figure 1: Existing QAT baselines such as EfficientQAT and Bitdistiller achieves near lossless performance on general benchmarks, but remaining a tremendous gap on downstream tasks at extremely low-bit. Our RoQAT significantly improves downstream reasoning capabilities while remaining comparable performance on general benchmarks

Motivated by these observations, we propose **RoQAT**, a robust QAT framework for improving downstream adaptation of extremely low-bit LLMs, as illustrated in Figure 2. RoQAT first introduces *sensitivity-adaptive progressive quantization* (SAPQ) to stabilize extremely low-bit training. Instead of following a uniform progression schedule, RoQAT estimates the block quantization sensitivity of each group along the actual progressive training direction and dynamically assigns group-specific progression rates, allowing more sensitive groups to approach the target precision more conservatively. Moreover, RoQAT incorporates *quantization-aware sharpness minimization* (QSAM) to improve the robustness and generalization of low-precision weights. It performs sharpness-aware optimization (Foret et al., 2020; Kwon et al., 2021) in the weight space during block-wise training and further extends it to the scaling space during end-to-end refinement, which encourages the optimization to converge toward flatter regions of the loss landscape. Together, these two components improve both the stability of low-bit training and the robustness of the quantized model, which better preserve downstream reasoning capabilities.

Extensive experiments on instruction-tuned LLMs demonstrate that our RoQAT significantly improves downstream reasoning capabilities compared with current strong QAT baselines or other downstream-specific methods while remaining performance on general benchmarks. For example, RoQAT achieves comparable general performance to EfficientQAT (61.62% vs. 61.38%) at 2-bit on Qwen3-4B, while improving the average downstream performance by 16.15%. Our RoQAT suggests a new QAT paradigm that shifts downstream capability preservation from separate post-quantization recovery into the QAT process itself. Our key contributions are summarized as follows:



(a) Sensitivity–adaptive Progressive Quantization (b) Quantization–aware sharpness minimization

Figure 2: Overview of RoQAT. (a) Sensitivity-adaptive progressive quantization assigns group-specific progression rates to allow sensitive groups training stably; (b) Quantization-aware sharpness minimization further improves downstream generalization by encouraging flatter loss landscapes.

- We reveal a significant gap between general capability recovery and downstream reasoning preservation in extremely low-bit QAT, and formulate training and progression instability and sharp loss landscapes as two important challenges.
- We propose RoQAT, which includes sensitivity-adaptive progressive quantization (SAPQ) to consider different quantization sensitivity during progressive training and quantization-aware sharpness minimization (QSAM) for improving robustness and downstream generalization under extremely low-bit QAT.
- Extensive experiments show that RoQAT significantly improves downstream reasoning performance while preserving general capabilities on instruction-tuned LLMs and hybrid attention LLMs compared with recent QAT and downstream-specific baselines.

2 RoQAT

In this section, we introduce **RoQAT**, a robust extremely low-bit QAT framework designed to preserve downstream capabilities. RoQAT improves QAT from two complementary aspects. First, sensitivity-adaptive progressive quantization (SAPQ) stabilizes extremely low-bit training by assigning different progression rates to quantization groups according to their sensitivity. Second, quantization-aware sharpness minimization (QSAM) improves the robustness and generalization of low-bit optimization by encouraging flatter loss landscapes. We build our RoQAT on the strong two-stage QAT framework EfficientQAT (Chen et al., 2025), which consists of block-wise weight training and end-to-end scale refinement. On top of this pipeline, RoQAT apply sensitivity-adaptive progressive quantization in the block-wise stage and sharpness-aware optimization in both stages.

2.1 Quantization-aware Training Preliminaries

We consider standard uniform affine quantization for model weights. Given a full-precision weight tensor W, we divide it into quantization groups $\{W_g\}_{g=1}^{G}$, where each group shares a scale s_g and a zero-point z_g. For a target bit-width b, the integer representation of the g-th group is obtained as:

$$W_g^{\text{int},b} = \text{clamp}\left(\left\lfloor \frac{W_g}{s_g} \right\rceil + z_g, 0, 2^b - 1\right), \tag{1}$$

where $\lfloor \cdot \rceil$ denotes the round-to-nearest operation. The quantized integer weights are then dequantized for forward computation:

$$\widehat{W}_g^{(b)} = s_g \left(W_g^{\text{int},b} - z_g\right), \tag{2}$$

where W_g denotes the full-precision trainable weights, while $\widehat{W}_g^{(b)}$ denotes the corresponding dequantized weights used in the forward pass. During QAT, the quantization and dequantization nodes are incorporated into the computation, which allows the model weights and quantization parameters to be optimized under low-bit forward computation.

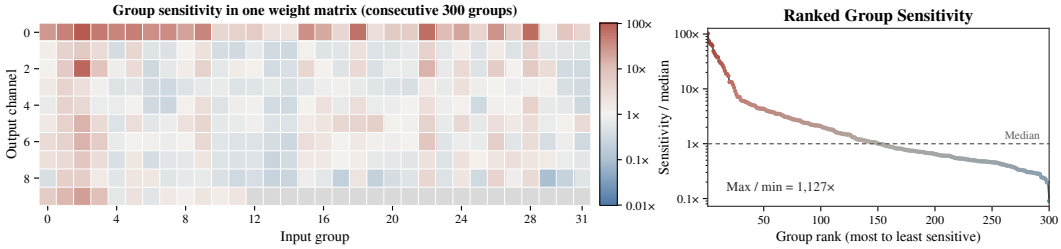


Figure 3: Group-wise quantization sensitivity visualization of one layer (1st group, self.o_proj) on Qwen3-4B at 2-bit. Sensitivity is measured by the block-output MSE between the full-precision and quantized groups. Different groups exhibit highly heterogeneous sensitivities, spanning over three orders of magnitude, which motivates group-specific progression rates.

2.2 Sensitivity-Adaptive Progressive Quantization

Directly quantizing a full-precision model to the target low bit-width at the beginning of QAT can introduce severe performance degradation, making the subsequent training unstable and difficult to optimize. Progressive quantization provides a natural solution by gradually exposing the model to increasing quantization difficulty. However, existing progressive schemes typically regulate this process with predefined or coarse-grained schedules (Tu et al., 2025; Wang et al., 2026; Xu et al., 2026), *e.g.*, stage-wise or tensor-wise progression, which overlooks the heterogeneous tolerance of different quantization groups.

To examine this issue, we empirically measure the quantization sensitivity of individual groups by the block-output reconstruction error when they are quantized to the target bit-width. As shown in Figure 3, even in the same weight matrix, different groups exhibit highly heterogeneous sensitivities across more than three orders of magnitude. This observation suggests that applying a uniform progression rate to all groups may be suboptimal. Sensitive groups should approach the target precision more conservatively, while insensitive groups can progress more aggressively.

Group-level progressive training Based on the observation above, we propose sensitivity-adaptive progressive quantization (SAPQ), where we perform progressive training at the group-level in the block-wise training stage. First, let $W_{s,g}$ and $W_{t,g}$ denote the dequantized weights of the g-th quantization group under the source and target bit-widths, respectively, where $b_s > b_t$. The actual progressive quantization direction is defined as:

$$\Delta W_g = W_{t,g} - W_{s,g}. \tag{3}$$

We use 8-bit as the source bit-width throughout our experiments. Instead of switching directly from $W_{s,g}$ to $W_{t,g}$, we introduce a group-specific progression rate $r_g \in [0, 1]$ and construct the effective weight used for forward computation as:

$$\widetilde{W}_g = W_{s,g} + r_g \Delta W_g. \tag{4}$$

As illustrated in Figure 2, when $r_g = 0$, the group remains at the source quantization state, while $r_g = 1$ corresponds to the target low-bit state. Intermediate values provide a continuous transition between the two states, which alleviates the abrupt disturbance caused by direct low-bit quantization.

Group-wise progression sensitivity The key question is how to efficiently determine r_g for different groups. Although the block-output reconstruction error used in Figure 3 provides a direct empirical measure, evaluating it for every group at each training step would introduce significant overheads. To address this issue, we use a first-order approximation to estimate the sensitivity online from the current optimization state. We denote the block-wise reconstruction loss as $\mathcal{L}_{\text{rec}}$ and the gradient with respect to the full-precision weight as $G = \nabla_W \mathcal{L}_{\text{rec}}$. For the g-th quantization group, we define its progression sensitivity as:

$$S_g = \left| \langle G_g, \Delta W_g \rangle \right|. \tag{5}$$

This approximation follows from the first-order Taylor expansion. When the g-th group moves by a small amount α along the progression direction, the corresponding change in the reconstruction

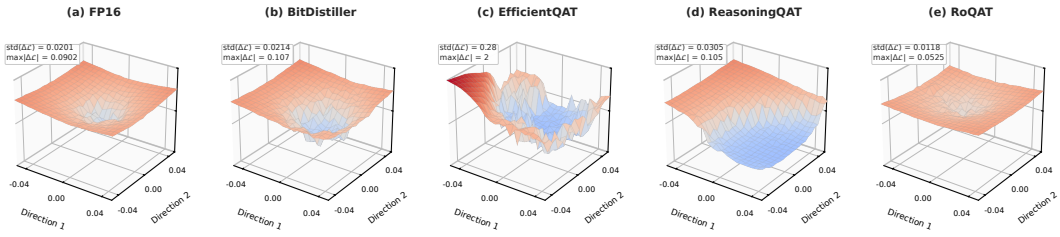


Figure 4: Loss landscape visualization of FP16 and different QAT methods on Qwen3-4B at 2-bit. RoQAT exhibits a much flatter landscape than existing QAT methods and is similar to FP16, which demonstrates our improved local robustness under extremely low-bit quantization.

objective can be approximated as:

$$\mathcal{L}_{\text{rec}}\left(W+\alpha\Delta W_g\right)-\mathcal{L}_{\text{rec}}(W)\approx\alpha\left\langle G_g,\Delta W_g\right\rangle. \tag{6}$$

Therefore, S_g serves as a local directional sensitivity at current progression stage. A large S_g indicates that progressing along ΔW_g is more likely to perturb the reconstruction objective, while a small S_g indicates stronger tolerance to the progression.

Sensitivity-adaptive progression Based on this sensitivity, we assign each group an adaptive target progression rate:

$$r_{g,\text{tar}}=\text{clip}\left(\frac{\lambda_p}{S_g+\epsilon},0,1\right), \tag{7}$$

where λ_p controls the overall progression strength, and ϵ is a small constant for numerical stability. Consequently, insensitive groups with small S_g can rapidly approach the target bit-width, while highly sensitive groups receive smaller progression rates and transition more conservatively.

Since the sensitivity estimated from an individual mini-batch may fluctuate during training, we further smooth the progression rate using an exponential moving average (EMA):

$$r_g^{(k+1)}=(1-\mu)r_g^{(k)}+\mu r_{g,\text{tar}}^{(k)}, \tag{8}$$

where μ controls the update rate. Since r_g is updated smoothly, the actual per-step movement is $\Delta r_g\Delta W_g$, which keeps the progression local even when the bit-width gap distance is large. In this way, RoQAT dynamically adapts the progression at the granularity of individual quantization groups, which provides a fine-grained and sensitivity-aware training process for extremely low-bit QAT. With this sensitivity-adaptive progression, the block-wise training avoids aggressive progression for sensitive groups and achieves a smoother convergence to the target bit-width.

2.3 Quantization-Aware Sharpness Minimization

Although progressive quantization stabilizes extremely low-bit training, the quantized model may still converge to sharp regions of the loss landscape. As illustrated in Figure 4, existing QAT methods exhibit noticeably sharper landscapes than the FP16 model, indicating stronger sensitivity to local parameter perturbations. Such sharpness can limit the robustness and generalization of the quantized model (Foret et al., 2020; Liu et al., 2021), which is particularly undesirable for preserving downstream reasoning capabilities.

Weight-space QSAM To alleviate this issue, we introduce quantization-aware sharpness minimization (QSAM) into the two-stage training. Firstly, in the block-wise QAT stage, the full-precision weight W is optimized, while the forward computation is performed using the progressively quantized weight defined in Eq. 4. Let $\mathbf{r}=\{r_g\}$ denote the progression rates of all quantization groups, and the block-wise reconstruction loss is formulated as:

$$\mathcal{L}_{\text{rec}}(W,\mathbf{r})=\left\|\mathcal{B}_{\text{FP}}(x)-\mathcal{B}_{\text{Q}}\big(x;\widetilde{W}(W,\mathbf{r})\big)\right\|_2^2, \tag{9}$$

where $\mathcal{B}_{\text{FP}}$ and $\mathcal{B}_{\text{Q}}$ denote the full-precision and quantized blocks, respectively. Instead of only minimizing the reconstruction loss at the current point, we optimize the worst-case loss in its local

neighborhood:

$$\min_{W} \max_{\epsilon_W \in \mathcal{N}_W} \mathcal{L}_{\text{rec}}\left(W + \epsilon_W, \mathbf{r}\right), \tag{10}$$

where ϵ_W denotes a local weight perturbation. Following (Kwon et al., 2021), $\mathcal{N}_W$ is constructed using weight-scale-adaptive perturbations to consider the different magnitude scales of weights. The perturbed weights are still evaluated through the progressively quantized forward process, such that the optimization explicitly penalizes sharp variations of the low-bit reconstruction objective. In this formulation, $\mathbf{r}$ controls how aggressively each group approaches the target precision, while ϵ_W regularizes the local sharpness around the corresponding progressive quantization state. Therefore, the block-wise stage simultaneously improves training stability and local weight robustness.

Scale-space QSAM The end-to-end stage freezes the model weights and globally refines the quantized model by optimizing its quantization parameters, primarily the scales s. Since the final model robustness depends not only on the quantized weights but also on the stability of these quantization parameters, we further apply QSAM in this space. Specifically, we optimize:

$$\min_{s} \max_{\epsilon_s \in \mathcal{N}_s} \mathcal{L}_{\text{E2E}}(s + \epsilon_s), \tag{11}$$

where $\mathcal{L}_{\text{E2E}}$ denotes the autoregressive language-modeling loss and ϵ_s is the adaptive perturbation applied to the quantization scales. Since the model weights are frozen, this objective directly encourages the end-to-end refinement to search for scale configurations whose language-modeling loss remains stable in a local neighborhood.

As shown in Figure 4, RoQAT significantly flattens the loss landscape and achieves a geometry close to the FP16 model with the help of two-space QSAM, which demonstrates improved local robustness and supports better downstream generalization under extremely low-bit quantization.

3 Experiments

3.1 Experimental Setup

Models and evaluations We conduct extensive experiments on Qwen3 (Team, 2026) instruction-tuned models (4B and 8B) and Qwen3.5 (Team, 2026) hybrid-attention model (4B) under 3/2-bit quantization (groupsize 128). To validate the effectiveness, we compare our RoQAT with two representative general-purpose QAT methods, EfficientQAT (Chen et al., 2025) and BitDistiller (Du et al., 2024), and the state-of-the-art downstream-purpose method ReasoningQAT (Okoshi et al., 2026). We further include two quantization-aware fine-tuning methods, ApiQ (Liao et al., 2024) and QA-LoRA (Xu et al., 2024), to compare against methods that recover model capabilities through parameter-efficient adaptation. In addition, two recent progressive QAT methods, Bit-by-bit (Xu et al., 2026) and HESTIA (Wang et al., 2026), are also included into comparison. All the baselines are reproduced by us following the same setup and environment to ensure the fairness. For evaluation, we consider both general capabilities and downstream reasoning capabilities. General performance is evaluated on ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2021) and TruthfulQA (Lin et al., 2022). For downstream evaluation, we use GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021) for mathematical reasoning, IFEval (Zhou et al., 2023) for instruction following, and HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) for code generation. The open-sourced toolkit lm-eval-harness (Gao et al., 2021) is used for evaluation.

Training details Followed by ReasoningQAT, we use mixed-domain data as the training dataset. Specifically, OpenThoughts-1.2M (Guha et al., 2026) and RedPajama-1T (Weber et al., 2024) are used in block-wise training and Deita-10K (Liu et al., 2024a) is used in end-to-end refinement. During block-wise training, we use 4096 samples with a context length of 2048 and optimize each Transformer block for 10 epochs with batchsize 2. The learning rates for weight and quantization parameters are set to 2×10^{-5} and 1×10^{-4}, respectively. During end-to-end refinement, we optimize the scales for 1000 steps with batchsize 16 and the learning rate 2×10^{-5}. For our SAPQ, we perform 8-bit RTN at the beginning (source bit-width). We set the progression strength λ_p and EMA parameter μ to 0.05 following (Morales-Brotons et al., 2024). For both weight and scale space QSAM, the perturbation radius ρ are all set to 0.05 following (Kwon et al., 2021). All experiments are conducted on NVIDIA A100-80GB GPUs.

Table 1: Comparison results of RoQAT and representative QAT methods on downstream and general tasks at 3/2-bit (g128) on Qwen3 instruction-tuned LLMs and Qwen3.5 hybrid attention LLMs.

Model	Bits	Method	**Downstream Benchmarks** (%, ↑)						General Benchmarks (%, ↑)						
			GSM8K	MATH	IFEval	HumE	MBPP	**Avg.**	AE	AC	HS	PQ	WG	TQ	**Avg.**
Qwen3-4B	–	FP16	85.14	53.20	37.77	70.73	61.80	61.73	78.20	54.10	68.43	74.97	65.75	54.85	66.05
	3	ApiQ	55.65	42.40	37.89	39.02	47.20	44.43	72.60	46.67	62.84	73.18	64.48	50.47	61.71
		QA-LoRA	66.72	34.20	35.25	39.63	48.40	44.84	**77.80**	51.45	65.57	75.46	67.01	48.80	64.35
		BitDistiller	77.33	42.20	31.29	43.90	47.60	48.46	74.41	50.09	64.59	74.32	65.82	53.11	63.72
		EfficientQAT	69.90	38.20	34.17	42.07	49.00	46.67	74.71	49.06	**68.44**	75.63	66.14	**54.41**	64.73
		ReasoningQAT	78.77	44.00	35.85	50.61	50.40	51.93	71.55	48.55	64.62	73.99	63.69	51.89	62.38
		RoQAT	**79.38**	**44.80**	**40.17**	**57.32**	**56.20**	**55.57**	77.23	**51.62**	67.89	**75.68**	**67.25**	52.23	**65.32**
	2	ApiQ	1.36	4.80	20.26	6.71	8.20	8.27	56.69	33.02	45.36	67.25	55.17	39.16	49.44
		QA-LoRA	2.27	3.60	31.65	11.59	8.20	11.46	63.05	39.93	50.67	69.26	58.64	46.28	54.64
		BitDistiller	30.55	12.20	25.06	13.41	21.00	20.44	60.52	37.97	50.42	67.95	60.14	46.92	53.99
		EfficientQAT	36.92	11.80	29.14	17.68	25.20	24.15	72.39	45.65	**63.20**	**73.61**	62.27	51.18	61.38
		ReasoningQAT	54.10	32.40	**38.01**	10.37	27.80	32.54	66.46	41.55	54.88	71.33	60.30	46.98	56.92
		RoQAT	**61.64**	**32.80**	36.69	**34.15**	**36.20**	**40.30**	**72.77**	**47.53**	61.18	72.74	**63.69**	**51.78**	**61.62**
Qwen3-8B	–	FP16	87.49	54.00	38.73	64.63	64.80	61.93	80.89	56.48	74.95	77.64	68.03	54.50	68.75
	3	ApiQ	75.89	42.20	34.77	48.78	52.20	50.77	77.61	51.88	69.27	76.71	66.38	53.09	65.82
		QA-LoRA	72.33	37.00	33.45	47.56	48.80	47.83	**80.47**	53.84	72.13	76.99	**70.96**	47.27	66.94
		BitDistiller	83.55	46.80	37.29	59.15	60.60	57.48	77.95	51.79	71.49	76.28	65.82	53.77	66.18
		EfficientQAT	76.42	41.00	34.89	47.56	53.20	50.61	79.59	55.97	71.38	**77.91**	69.22	53.19	67.88
		ReasoningQAT	85.75	**50.00**	41.97	53.05	61.60	58.47	78.24	52.22	71.50	76.39	68.11	53.42	66.65
		RoQAT	**86.35**	49.80	**45.68**	**59.15**	**61.60**	**60.52**	79.88	**56.06**	**74.99**	77.26	69.38	**53.83**	**68.57**
	2	ApiQ	7.88	7.80	22.42	10.98	16.80	13.18	61.95	35.92	52.49	70.51	58.96	42.26	53.68
		QA-LoRA	0.00	5.60	33.33	10.37	15.60	12.98	30.68	26.19	51.81	55.06	60.46	46.50	45.12
		BitDistiller	42.91	16.20	25.18	7.32	26.60	23.64	67.80	41.30	59.41	71.11	61.01	50.49	58.52
		EfficientQAT	54.21	18.80	33.21	15.24	31.00	30.49	**77.15**	51.11	**70.71**	**75.14**	66.54	**52.10**	**65.46**
		ReasoningQAT	72.63	35.60	28.18	21.34	36.20	38.79	71.42	45.73	63.58	73.67	62.35	48.41	60.86
		RoQAT	**74.43**	**43.00**	**36.81**	**32.93**	**44.80**	**46.39**	76.47	**51.37**	69.02	74.92	**68.11**	51.09	65.16
Qwen3.5-4B	–	FP16	74.07	46.60	44.12	58.54	39.20	52.51	75.51	54.35	73.16	78.29	69.93	48.85	66.68
	3	ApiQ	64.75	41.40	44.48	35.37	27.20	42.64	73.74	49.74	67.90	76.17	66.22	47.00	63.46
		QA-LoRA	57.92	30.60	**56.95**	41.46	39.80	45.35	**79.97**	**55.89**	69.47	76.55	66.06	46.84	65.80
		BitDistiller	71.11	42.80	42.33	39.63	38.20	46.81	75.97	52.22	69.79	76.77	66.85	49.03	65.11
		EfficientQAT	69.75	34.40	48.80	31.71	42.20	45.37	78.54	55.55	**72.54**	**78.40**	**69.14**	**54.65**	**68.14**
		ReasoningQAT	**78.09**	37.60	33.21	41.46	28.60	43.79	71.59	50.00	68.72	75.30	67.25	49.30	63.69
		RoQAT	74.00	**44.80**	52.16	**44.51**	**45.20**	**52.13**	79.17	54.78	72.40	**78.40**	68.90	54.21	67.98
	2	ApiQ	5.23	5.80	31.89	4.88	1.80	9.92	58.63	34.39	50.28	69.26	58.72	44.95	52.71
		QA-LoRA	4.85	5.60	36.09	15.24	12.20	14.80	65.78	40.61	53.53	69.21	59.83	46.58	55.92
		BitDistiller	24.49	11.20	36.93	10.98	16.40	20.00	65.57	40.61	55.17	69.97	59.75	44.76	55.97
		EfficientQAT	30.93	8.80	40.53	15.24	19.60	23.02	**70.83**	**45.90**	65.92	73.88	63.61	50.43	61.76
		ReasoningQAT	59.67	29.20	29.86	10.37	9.60	27.74	59.97	37.88	58.63	71.60	63.06	48.28	56.57
		RoQAT	**60.50**	**30.80**	**42.69**	**21.95**	**30.40**	**37.27**	70.41	43.34	**66.07**	**74.54**	**66.22**	**52.36**	**62.16**

3.2 Main Results on Downstream and General Benchmarks

Downstream benchmarks. Table 1 shows that RoQAT consistently achieves the strongest downstream performance across different models and target bit-widths. On Qwen3-4B, RoQAT improves the average downstream performance over the strongest baseline by 3.64% and 7.76% at 3-bit and 2-bit, respectively. Similar improvements are observed on Qwen3-8B, where RoQAT achieves gains of 2.05% and 7.60%. On Qwen3.5-4B, RoQAT further outperforms the strongest baseline by 5.32% at 3-bit and 9.53% at 2-bit. Notably, the improvement is consistently larger under 2-bit quantization, where existing methods suffer more severe degradation. These results demonstrate that RoQAT is particularly effective at preserving downstream capabilities under extremely low-bit quantization.

General benchmarks. Despite its substantial improvements on downstream tasks, RoQAT maintains comparable performance on general benchmarks. In particular, RoQAT reaches 65.32% and 61.62% on Qwen3-4B at 3-bit and 2-bit, respectively, which outperforms all other quantized methods. On Qwen3-8B and Qwen3.5-4B, RoQAT performs either the best or only 0.30% below the strongest baseline. This indicates that RoQAT preserves a strong balance between downstream reasoning and general performance across different models and bit-widths.

Table 2: Comparison with progressive QAT methods under 2-bit quantization. RoQAT is evaluated without QSAM to isolate the effect of sensitivity-adaptive progression.

Model	Method	**Downstream Benchmarks** (%, ↑)						General Benchmarks (%, ↑)						
		GSM8K	MATH	IFEval	HumE	MBPP	**Avg.**	AE	AC	HS	PQ	WG	TQ	**Avg.**
Qwen3-4B	Bit-by-Bit	44.58	30.20	**33.09**	17.07	30.80	31.15	65.78	41.98	54.70	**72.31**	62.19	**49.38**	57.72
	HESTIA	23.65	17.40	23.50	13.41	13.00	18.19	58.25	34.39	49.53	68.93	56.59	44.71	52.07
	RoQAT (w/o QSAM)	**60.50**	**37.00**	29.50	**28.05**	**31.60**	**37.33**	**70.20**	**44.11**	**56.11**	71.44	**62.75**	47.52	**58.69**
Qwen3.5-4B	Bit-by-Bit	45.26	27.80	**39.93**	21.95	19.80	30.95	**66.37**	40.19	61.01	72.80	62.27	46.88	58.25
	HESTIA	30.93	10.40	31.18	12.80	4.20	17.90	57.53	35.07	55.05	71.60	61.64	**49.53**	55.07
	RoQAT (w/o QSAM)	**47.23**	**29.40**	36.33	**27.44**	**25.00**	**33.08**	65.45	**40.78**	**61.03**	**73.12**	**64.88**	48.25	**58.92**

3.3 Comparison with Progressive QAT Methods

We further compare RoQAT with representative progressive QAT methods, including Bit-by-Bit and HESTIA, to examine the effectiveness of our SAPQ. For a fair comparison, all methods use the same training dataset (Okoshi et al., 2026) and are evaluated under block-wise training. We remove QSAM in RoQAT in this experiment to isolate the contribution of our progression strategy.

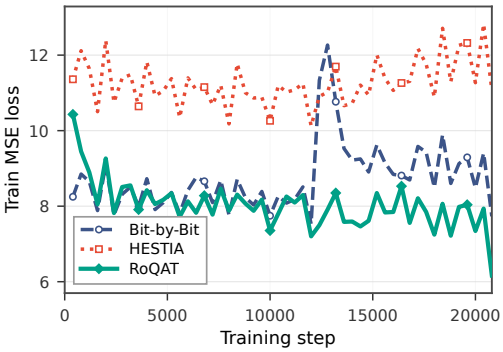


Figure 5: Block-wise training loss comparison of different progressive QAT methods on Qwen3-4B at 2-bit. RoQAT achieves a more stable progression and the lowest final loss.

As shown in Table 2, RoQAT consistently achieves better downstream performance than existing progressive QAT methods. On Qwen3-4B, RoQAT outperforms the strongest baseline, Bit-by-Bit, by 6.18%. On Qwen3.5-4B, RoQAT reaches 33.08%, while Bit-by-Bit is 30.95%. Meanwhile, RoQAT also maintains strong general capabilities, which achieves the highest average general performance on both models. These results show that adapting the progression at the group-level is more effective for preserving downstream capabilities.

We further examine the block-wise training loss. As shown in Figure 5 which records the data every 400 training steps, RoQAT maintains a relatively stable loss trajectory while gradually approaching the target 2-bit precision, and reaches the lowest loss at last. In contrast, Bit-by-Bit exhibits a clear loss spike during its discrete bit-width transition from 8-bit to 4-bit and then to 2-bit, which reflects the abrupt change in quantization difficulty. HESTIA maintains a noticeably higher reconstruction loss throughout most of the training process. Overall, the results suggest that our SAPQ can provide a smoother and more stable progression to the target bit-width during training.

3.4 Ablation Study

Table 3 examines the contribution of SAPQ and the two-space QSAM design on Qwen3-4B at 2-bit. SAPQ alone achieves 59.21% on GSM8K and 33.54% on HumanEval, which demonstrates the benefit of stabilizing the progressive training process. Applying QSAM only to the scale-space during end-to-end refinement further improves GSM8K to 60.73%. And the full RoQAT, which additionally performs weight-space QSAM during block-wise training, achieves the best performance on both benchmarks with 61.64% on GSM8K and 34.15% on HumanEval. These results indicate that SAPQ and two-space QSAM are complementary, and jointly improving the training trajectory and local robustness brings more consistent downstream performance.

Table 3: Ablation study of RoQAT on Qwen3-4B at 2-bit. S-QSAM and W-QSAM denote scale and weight space QSAM, respectively.

SAPQ	**S-QSAM**	**W-QSAM**	GSM8K	HumE
–	–	–	55.42	29.88
✓	–	–	59.21	33.54
✓	✓	–	60.73	31.10
✓	✓	✓	**61.64**	**34.15**

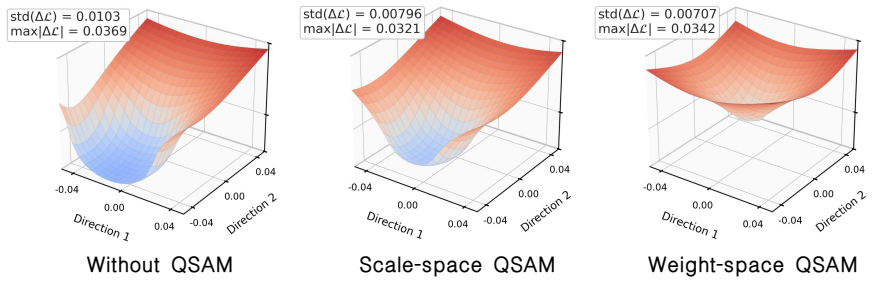


Figure 6: Ablation of QSAM on the loss landscape of Qwen3-4B at 2-bit. Both weight-space QSAM and scale-space QSAM produce flatter local landscapes.

Table 4: Performance and training-cost comparison with DAPO on Qwen3-4B. RoQAT better preserves both downstream and general capabilities with lower training cost.

Bits	Method	Downstream Benchmarks (%, ↑)						General Avg. (%, ↑)	GPU-hours ↓
		GSM8K	MATH	IFEval	HumE	MBPP	**Avg.**		
3	DAPO	**80.21**	**46.00**	33.81	32.93	46.80	47.95	60.86	106.00
	RoQAT	79.38	44.80	**40.17**	**57.32**	**56.20**	**55.57**	**65.32**	**30.50**
2	DAPO	12.13	9.20	25.42	0.00	6.80	10.71	43.89	85.46
	RoQAT	**61.64**	**32.80**	**36.69**	**34.15**	**36.20**	**40.30**	**61.62**	**18.05**

We further visualize the loss landscape in Figure 6 to examine the geometric effect of QSAM. Without QSAM, the quantized model exhibits a relatively sharper local landscape. Applying S-QSAM during end-to-end refinement reduces the local loss variation in the scale space, while Weight-space QSAM during block-wise training produces a further flatter landscape under local perturbations. These results show that sharpness-aware optimization in both stages encourages flatter local loss landscapes and improves the robustness of extremely low-bit models to parameter perturbations.

3.5 Efficiency and Performance Comparison with Reinforcement Learning

An alternative way to recover capabilities degraded by quantization is additional reinforcement-learning post-training. To examine this alternative, we compare RoQAT with DAPO (Yu et al., 2026) on Qwen3-4B. As shown in Table 4, although DAPO retains strong performance on some reasoning tasks at 3-bit, its overall downstream and general performance drops to 47.95% and 60.86%, while RoQAT maintains 55.57% and 65.32%, respectively. The gap becomes larger at 2-bit, where DAPO achieves only 10.71% downstream and 43.89% general performance, compared with 40.30% and 61.62% for RoQAT. Meanwhile, RoQAT requires much fewer GPU-hours at both bit-widths. These results suggest that directly preserving model capabilities during QAT is more effective and efficient than recovering them through additional reinforcement learning after quantization.

4 Conclusion

In this paper, we investigate the tremendous gap between general capability recovery and downstream reasoning preservation in extremely low-bit QAT. We identify two important challenges, where existing progressive quantization schemes overlook sensitivity differences during low-bit adaptation, while extremely low-bit QAT may converge to sharp optimization landscapes. To address these challenges, we propose RoQAT, which includes sensitivity-adaptive progressive quantization (SAPQ) and quantization-aware sharpness minimization (QSAM). SAPQ performs group-level progressive training according to quantization sensitivity, while QSAM improves the robustness and generalization of the quantized model by encouraging flatter loss landscapes in both weight and scale spaces. Extensive experiments on Qwen3 instruction-tuned and Qwen3.5 hybrid-attention LLMs demonstrate that RoQAT consistently improves downstream reasoning performance under extremely low-bit quantization while maintaining comparable general capabilities.

AI USE STATEMENT

In this work, we used generative AI tools primarily for language polishing, including improving grammar, clarity, readability and academic expression of the manuscript. The research methodology, technical derivations, experimental design, implementation, analysis and conclusions were developed and verified by the authors. All AI-assisted text was carefully reviewed, revised and repeatedly checked by the authors to ensure technical correctness, consistency with the conducted experiments and faithful representation of our contributions. We take full responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper focuses on improving the efficiency and robustness of extremely low-bit quantization for large language models. Our experiments are conducted on publicly available models, datasets, and evaluation benchmarks, and do not involve human subjects, user studies or the collection of private or sensitive personal information. The proposed method is intended to reduce the memory and computational costs of deploying large language models. As with other model compression techniques, improved deployment efficiency may also make existing capabilities of language models more accessible. Therefore, potential risks associated with the underlying models remain relevant. We do not introduce new data collection procedures or capabilities beyond those of the original models. We take responsibility for the ethical considerations associated with the methodology, experiments and reported results.

REPRODUCIBILITY STATEMENT

We provide detailed model configurations, training hyperparameters, data composition and evaluation methods in the content and appendix. All models, training datasets, and evaluation benchmarks used in our experiments are publicly available and properly cited. Baseline methods are reproduced following their official implementations and configurations whenever available, under consistent quantization and evaluation settings. We will release RoQAT and the training/evaluation scripts to help reproduction of the reported results soon.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7360–7371, 2022.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 102–116, 2024.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

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

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. A framework for few-shot language model evaluation. *Zenodo*, 2021.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. In *International Conference on Learning Representations*, volume 2026, pp. 108059–108130, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International conference on machine learning*, pp. 5905–5914. PMLR, 2021.

Jiedong Lang, Zhehao Guo, and Shuyu Huang. A comprehensive study on quantization techniques for large language models. In *2024 4th International conference on artificial intelligence, robotics, and communication (ICAIRC)*, pp. 224–231. IEEE, 2024.

Jung Hyun Lee, Seungjae Shin, Vinnam Kim, Jaeseong You, and An Chen. Unifying block-wise ptq and distillation-based qat for progressive quantization toward 2-bit instruction-tuned llms. *arXiv preprint arXiv:2506.09104*, 2025.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pp. 4582–4597, 2021.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021.

Zhen Li, Yupeng Su, Runming Yang, Congkai Xie, Zheng Wang, Zhongwei Xie, Ngai Wong, and Hongxia Yang. Quantization meets reasoning: Exploring llm low-bit quantization degradation for mathematical reasoning. *arXiv preprint arXiv:2501.03035*, 2025.

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

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.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Jing Liu, Jianfei Cai, and Bohan Zhuang. Sharpness-aware quantization for deep neural networks. *arXiv preprint arXiv:2111.12273*, 2021.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. Quantization hurts reasoning? an empirical study on quantized reasoning models. *arXiv preprint arXiv:2504.04823*, 2025.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *International Conference on Learning Representations*, volume 2024, pp. 22353–22373, 2024a.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the association for computational linguistics: ACL 2024*, pp. 467–484, 2024b.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024.

Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *arXiv preprint arXiv:2411.18704*, 2024.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International conference on machine learning*, pp. 7197–7206. PMLR, 2020.

Yasuyuki Okoshi, Hikari Otsuka, Daichi Fujiki, and Masato Motomura. Towards quantization-aware training for ultra-low-bit reasoning LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Azsd2qyK6C.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Zhijun Tu, Jian Li, Yuanyuan Xi, Siqi Liu, Chuanjian Liu, Hanting Chen, Jie Hu, and Yunhe Wang. Rethinking 1-bit optimization leveraging pre-trained large language models. *arXiv preprint arXiv:2508.06974*, 2025.

Guoan Wang, Feiyu Wang, Zongwei Lv, Yikun Zong, and Tong Yang. Hestia: A hessian-guided differentiable quantization-aware training framework for extremely low-bit llms. *arXiv preprint arXiv:2601.20745*, 2026.

Jianxun Wang and Yixiang Chen. A review on code generation with llms: Application and evaluation. In *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, pp. 284–289. IEEE, 2023.

Zheng Wang, Juncheng B Li, Shuhui Qu, Florian Metze, and Emma Strubell. Squat: Sharpness-and quantization-aware training for bert. *arXiv preprint arXiv:2210.07171*, 2022.

Maurice Weber, Daniel Y Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37: 116462–116492, 2024.

Binxing Xu, Hao Gu, Lujun Li, Hao Wang, Bei Liu, Jiacheng Liu, Qiyuan Zhu, Xintong Yang, Chao Li, Sirui Han, et al. Bit-by-bit: Progressive qat strategy with outlier channel splitting for stable low-bit llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22284–22299, 2026.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 52401–52418, 2024.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. Turboquant: Online vector quantization with near-optimal distortion rate. In *International Conference on Learning Representations*, volume 2026, pp. 56418–56439, 2026.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

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

A Related Works

A.1 Quantization-aware Training

Quantization-aware training (QAT) simulates low-bit computation during training to mitigate quantization-induced degradation, so that achieving much better performance than PTQ. Early work such as LSQ (Esser et al., 2019) introduces straight-through estimation to learn quantization step sizes and model weights, establishing an effective paradigm for standard QAT. LLM-QAT (Liu et al., 2024b) introduces data-free distillation for low-bit training, while BitNet b1.58 (Ma et al., 2024) demonstrates the potential of native ternary-weight training. More recently, BitDistiller (Du et al., 2024) combines QAT with self-distillation, and EfficientQAT (Chen et al., 2025) improves training efficiency through block-wise weight optimization and end-to-end scale refinement. Despite strong recovery on general benchmarks, these methods still suffer severe degradation on complex downstream reasoning tasks. ReasoningQAT (Okoshi et al., 2026) further targets this problem by constructing a downstream-oriented QAT dataset and introducing a reinforcement-like training objective. Nevertheless, preserving reasoning capabilities under extremely low precision remains quite challenging under aggressive quantization, particularly at 2-bit precision. In this paper, we improve extremely low-bit QAT from both the training trajectory and the robustness of the optimization landscape to address this limitation.

A.2 Progressive Quantization

Progressive quantization has been explored to stabilize extremely low-bit training by gradually increasing quantization difficulty. BinaryLLM (Tu et al., 2025) progressively transforms full-precision weights to binary representations, while UPQ (Lee et al., 2025) and Bit-by-Bit (Xu et al., 2026) adopt staged precision reduction to ease low-bit adaptation. HESTIA (Wang et al., 2026) follows a different route by progressively hardening a differentiable soft quantizer, with tensor-wise Hessian sensitivity controlling the annealing rate. Despite their effectiveness, existing methods mainly regulate progression at the global, stage-wise, or tensor level, which ignores the sensitivity heterogeneity across quantization groups. In contrast, we directly model the progression at the quantization-group level and assign group-specific progression rates according to their reconstruction sensitivity.

A.3 Sharpness-aware Minimization

Sharpness-aware minimization (SAM) (Foret et al., 2020) improves model generalization by jointly minimizing the training loss and the worst-case loss in a local parameter neighborhood, encouraging convergence toward flatter regions of the loss landscape. ASAM (Kwon et al., 2021) further introduces adaptive perturbations to obtain a scale-invariant sharpness measure. The effectiveness of SAM has also been demonstrated for improving the generalization of language models (Bahri et al., 2022). Recent studies have explored sharpness-aware optimization in quantization. SAQ (Liu et al., 2021) studies how SAM perturbations should be constructed under quantization, and SQuAT (Wang et al., 2022) encourages flatter minima for low-bit BERT models. In this work, we employ sharpness-aware optimization in extremely low-bit LLM QAT to improve both the robustness and generalization of low-precision models, which operates over the weight space during block-wise training and the scale space during end-to-end refinement.

B More Comparison and Visualizations on RoQAT

B.1 More Group Sensitivity Visualizations

To further examine whether the group-wise sensitivity heterogeneity observed in Figure 3 is specific to a particular weight matrix, we provide more visualizations across different layers on Qwen3.5-4B. For each matrix, we still rank 300 consecutive quantization groups according to their block-reconstruction sensitivity and normalize the sensitivity by the median value. As shown in Figure 7, there can be seen clearly different quantization sensitivities, where the max-to-min sensitivity ratio ranges from tens to more than three orders of magnitude across the selected matrices. These results further support our motivation that using the same progression schedule for all quantization groups may be suboptimal and the progressive training should adapt to the sensitivity of different groups.

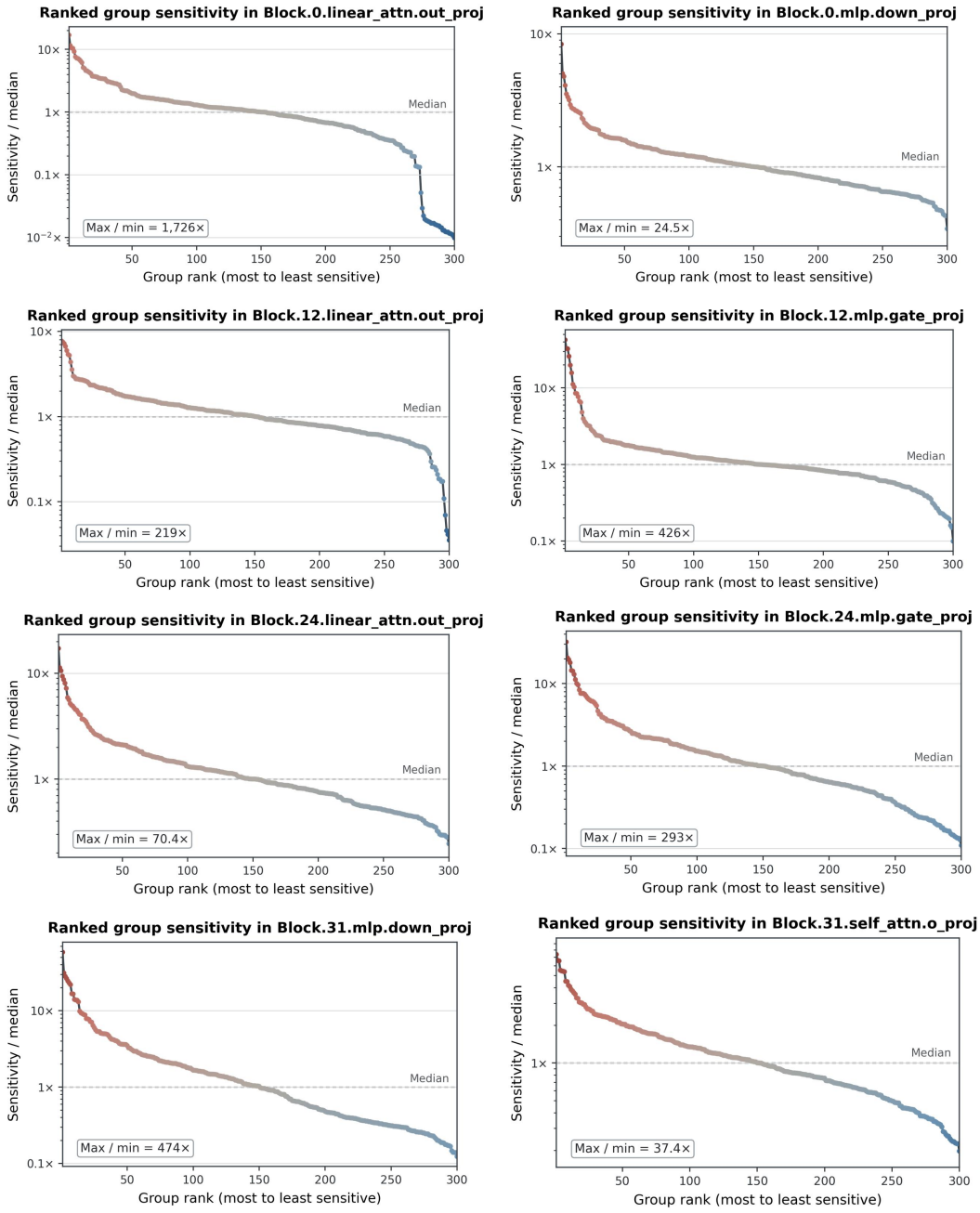
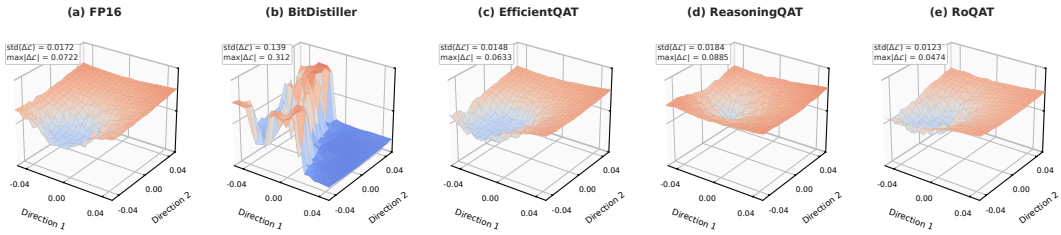


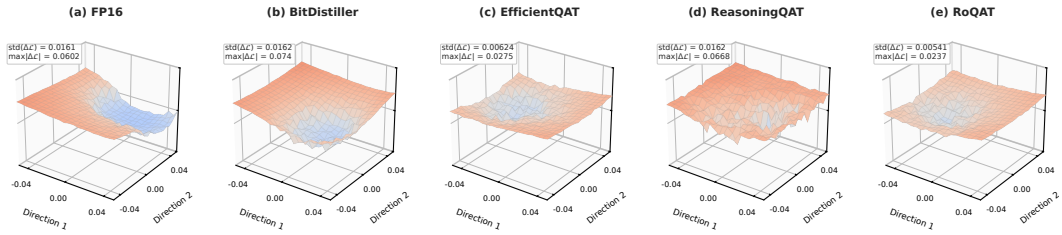
Figure 7: More group-sensitivity visualizations across different layers on Qwen3.5-4B.

B.2 More Loss Landscape Visualizations

In addition to Qwen3-4B in Figure 4, we provide more loss landscape visualizations on Qwen3-8B and Qwen3.5-4B. As shown in Figure 8, Across both models, RoQAT consistently exhibits the flattest loss landscape among the compared quantized methods. These results further prove that quantization-aware sharpness minimization improves the local robustness of extremely low-bit models, which supports the preservation of downstream capabilities.



(a) Qwen3-8B



(b) Qwen3.5-4B

Figure 8: Additional loss landscape visualizations on (a) Qwen3-8B and (b) Qwen3.5-4B at 2-bit. RoQAT consistently exhibits the flattest local loss landscape among the compared methods.

Table 5: Performance comparison using the same ReasoningQAT training dataset on Qwen3-4B and Qwen3.5-4B at 2-bit (g128).

Model	Method	**Downstream Benchmarks** (%, ↑)						General Benchmarks (%, ↑)						
		GSM8K	MATH	IFEval	HumE	MBPP	**Avg.**	AE	AC	HS	PQ	WG	TQ	**Avg.**
Qwen3-4B	ApiQ	10.92	13.60	26.50	6.10	6.00	12.62	57.32	33.36	46.25	66.21	58.48	43.37	50.83
	QA-LoRA	21.76	6.80	36.09	19.51	23.60	21.55	64.77	39.85	56.31	69.70	59.91	50.86	56.90
	BitDistiller	26.54	11.20	23.38	18.29	25.40	20.96	63.05	38.31	52.11	68.23	57.14	48.02	54.48
	EfficientQAT	57.92	**35.60**	36.57	28.05	36.00	38.83	**73.36**	45.22	**62.03**	**73.78**	63.30	48.89	61.10
	ReasoningQAT	55.34	29.40	**37.77**	28.05	31.80	36.47	73.11	43.60	60.83	73.01	61.96	50.64	60.53
	RoQAT	**61.64**	32.80	36.69	**34.15**	**36.20**	**40.30**	72.77	**47.53**	61.18	72.74	**63.69**	**51.78**	**61.62**
Qwen3.5-4B	ApiQ	10.24	7.80	33.21	4.88	6.60	12.55	54.88	34.04	50.45	68.72	57.85	47.25	52.20
	QA-LoRA	24.72	7.80	38.73	18.29	18.60	21.63	70.24	42.75	60.53	71.44	62.75	47.77	59.25
	BitDistiller	26.46	10.00	32.97	13.41	19.00	20.37	69.95	43.26	57.64	71.33	60.46	48.24	58.48
	EfficientQAT	57.32	29.40	**43.17**	21.95	28.00	35.97	**71.30**	**44.97**	**66.15**	**74.97**	64.88	50.96	**62.21**
	ReasoningQAT	49.81	28.80	39.21	21.34	25.60	32.95	69.74	43.86	64.13	74.59	65.04	49.76	61.19
	RoQAT	**60.73**	**30.80**	42.69	**26.22**	**30.40**	**38.17**	70.41	43.34	66.07	74.54	**66.22**	**52.36**	62.16

B.3 Comparison Results under ReasoningQAT Training Datasets

To further exclude the influence of training data, we compare different QAT methods under the same training data used by ReasoningQAT, while keeping their original optimization strategies unchanged. The results are reported in Table 5. We can observe that RoQAT consistently achieves the highest average downstream performance on both models. On Qwen3-4B, RoQAT reaches 40.30%, which outperforms BitDistiller by 19.34% and ReasoningQAT by 3.83%. On Qwen3.5-4B, RoQAT achieves 38.17%, compared with 20.37% for BitDistiller and 32.95% for ReasoningQAT. In particular, despite using the same training data as ReasoningQAT, RoQAT improves its downstream average by 3.83% and 5.22% on Qwen3-4B and Qwen3.5-4B, respectively. RoQAT also maintains strong general capabilities, which achieves the best or comparable average general performance on both models. These results indicate that RoQAT still demonstrate its downstream advantage under the same training dataset, which highlights the effectiveness of the proposed optimization strategy.

Table 6: Sensitivity analysis of the progression strength λ_p in SAPQ on Qwen3-4B and Qwen3.5-4B under 2-bit quantization (g128). All results are obtained after block-wise training without end-to-end refinement or QSAM. † denotes the default setting used in our main experiments.

Model	λ_p	**Downstream Benchmarks** (%, ↑)						**General Avg.** (%, ↑)
		GSM8K	MATH	IFEval	HumE	MBPP	**Avg.**	
Qwen3-4B	0.01	36.09	36.60	32.97	20.12	30.20	31.20	58.52
	0.05†	52.69	35.60	30.22	20.12	31.40	34.01	58.22
	0.10	56.48	35.40	31.29	23.17	30.40	35.35	58.35
	0.20	57.01	34.40	31.41	23.78	32.20	35.76	58.21
	0.30	51.33	37.60	34.89	18.29	31.60	34.74	58.56
Qwen3.5-4B	0.01	48.07	27.60	38.61	28.05	22.80	33.02	58.82
	0.05†	47.23	29.40	36.33	27.44	25.00	33.08	58.92
	0.10	50.87	26.60	36.45	25.00	24.00	32.58	58.15
	0.20	46.85	26.00	34.53	28.05	24.40	31.97	58.69
	0.30	51.25	29.60	38.61	23.17	22.20	32.97	58.65

Table 7: Sensitivity analysis of the EMA coefficient μ in SAPQ on Qwen3-4B and Qwen3.5-4B under 2-bit quantization (g128). All results are obtained after block-wise training without end-to-end refinement or QSAM. † denotes the default setting used in our main experiments.

Model	μ	**Downstream Benchmarks** (%, ↑)						**General Avg.** (%, ↑)
		GSM8K	MATH	IFEval	HumE	MBPP	**Avg.**	
Qwen3-4B	0.01	51.40	36.80	31.65	23.17	30.00	34.60	58.79
	0.025	43.29	36.40	33.45	25.00	33.80	34.39	58.36
	0.05†	52.69	35.60	30.22	20.12	31.40	34.01	58.22
	0.10	54.28	34.20	30.34	17.07	33.80	33.94	58.14
	0.20	27.75	36.80	29.38	23.78	33.60	30.26	58.60
Qwen3.5-4B	0.01	50.04	27.60	38.85	31.10	16.80	32.88	58.34
	0.025	46.25	27.00	33.93	26.22	21.40	30.96	57.87
	0.05†	47.23	29.40	36.33	27.44	25.00	33.08	58.92
	0.10	44.81	24.20	35.13	20.12	19.20	28.69	57.87
	0.20	43.90	27.00	35.25	28.05	20.60	30.96	58.74

C Hyper-parameter Analysis

C.1 Sensitivity to Progression Strength λ_p

We study the sensitivity of SAPQ to the progression strength λ_p on Qwen3-4B and Qwen3.5-4B at 2-bit. To isolate the effect of λ_p, we only perform block-wise training without end-to-end refinement or QSAM. As shown in Table 6, the performance does not vary monotonically with λ_p, and different models and tasks favor different progression strengths. In particular, except for $\lambda_p = 0.01$ on Qwen3-4B, SAPQ maintains relatively stable downstream performance over a broad range of λ_p. The general benchmark averages are also stable across all settings. Therefore, we use $\lambda_p = 0.05$ as a unified default for all models rather than tuning it separately for each model.

C.2 Sensitivity to EMA Coefficient μ

We further analyze the sensitivity of SAPQ to the EMA coefficient μ, which controls how rapidly the group-specific progression rate follows its sensitivity-adaptive target. To isolate its effect, we only perform block-wise training without end-to-end refinement or QSAM. As shown in Table 7, the

Table 8: Sensitivity analysis of the QSAM perturbation radius ρ on Qwen3-4B and Qwen3.5-4B under 2-bit quantization (g128). We evaluate ρ separately for weight-space QSAM during block-wise training and scale-space QSAM during end-to-end refinement. † denotes the default setting used in our main experiments.

Stage	Model	ρ	**Downstream Benchmarks** (%, ↑)						**General Avg.** (%, ↑)
			GSM8K	MATH	IFEval	HumE	MBPP	**Avg.**	
Weight-space QSAM	Qwen3-4B	0.01	47.92	37.20	31.77	28.05	33.80	35.75	58.22
		0.025	49.89	37.40	33.81	20.12	32.60	34.76	57.95
		0.05†	53.90	34.40	28.42	26.83	34.20	35.55	58.63
		0.10	35.94	36.00	32.13	18.90	32.40	31.07	57.44
		0.20	45.03	33.40	32.49	21.34	28.00	32.05	57.00
	Qwen3.5-4B	0.01	49.51	25.40	37.77	22.56	14.60	29.97	58.61
		0.025	47.38	24.40	33.57	21.95	17.20	28.90	58.30
		0.05†	47.01	25.00	38.49	23.78	19.20	30.70	58.64
		0.10	42.61	25.20	35.49	23.17	9.00	27.09	58.89
		0.20	50.34	24.80	34.89	18.90	21.20	30.03	58.45
Scale-space QSAM	Qwen3-4B	0.01	57.39	33.20	38.61	29.27	38.00	39.29	61.27
		0.05†	61.64	32.80	36.69	34.15	36.20	40.30	61.62
		0.10	58.30	31.60	37.77	29.88	36.20	38.75	61.44
		0.20	58.45	32.20	37.29	29.27	36.80	38.80	61.19
		0.50	56.79	31.80	36.21	29.88	36.80	38.30	61.15
	Qwen3.5-4B	0.01	60.35	31.40	43.88	21.95	30.40	37.60	62.20
		0.05†	60.50	30.80	42.69	21.95	30.40	37.27	62.16
		0.10	59.59	32.00	43.41	23.17	30.00	37.63	62.19
		0.20	59.67	31.00	43.17	23.17	29.80	37.36	62.12
		0.50	58.76	31.40	43.29	26.22	30.40	38.01	62.24

performance varies non-monotonically with μ, and different tasks exhibit different preferences. The general benchmark averages remain relatively stable across a broad range of μ, while larger values may cause more noticeable fluctuations on downstream tasks. Since μ mainly controls the trade-off between responsiveness and smoothing of the progression rate, we use $\mu = 0.05$ as a unified setting for all models rather than tuning it separately for individual tasks or architectures.

C.3 Sensitivity to QSAM Perturbation Radius ρ

We next study the sensitivity of QSAM to the perturbation radius ρ in both training stages. We evaluate weight-space QSAM during block-wise training and scale-space QSAM during end-to-end refinement separately, so that the effect of ρ in each parameter space can be examined independently. The results are summarized in Table 8. For weight-space QSAM, moderate perturbation generally provide stronger downstream performance, while excessively large perturbations can degrade the optimization, particularly on Qwen3-4B. The default $\rho = 0.05$ achieves competitive performance on Qwen3-4B and the highest downstream average on Qwen3.5-4B. For scale-space QSAM, the results are more stable across a wide range of ρ, especially on general benchmarks. On Qwen3-4B, $\rho = 0.05$ achieves the highest downstream average, while different values perform similarly on Qwen3.5-4B. Overall, no single alternative radius consistently dominates across models and training stages. Therefore, we use $\rho = 0.05$ followed by (Kwon et al., 2021) for both weight-space and scale-space QSAM throughout the main experiments.