# What Makes Low-Bit Quantization-Aware Training Work for Reasoning LLMs? A Systematic Study

**Anonymous ACL submission**


## Abstract

Reasoning models excel at complex tasks such as coding and mathematics, yet their inference is often slow and token-inefficient. To improve the inference efficiency, post-training quantization (PTQ) usually comes with the cost of large accuracy drops, especially for reasoning tasks under low-bit settings. In this study, we present a systematic empirical study of quantization-aware training (QAT) for reasoning models. Our key findings include: (1) Knowledge distillation is a robust objective for reasoning models trained via either supervised fine-tuning or reinforcement learning; (2) PTQ provides a strong initialization for QAT, improving accuracy while reducing training cost; (3) Reinforcement learning remains feasible for quantized models given a viable cold start and yields additional gains; and (4) Aligning the PTQ calibration domain with the QAT training domain accelerates convergence and often improves the final accuracy. Finally, we consolidate these findings into an optimized workflow (Reasoning-QAT), and show that it consistently outperforms state-of-the-art PTQ methods across multiple LLM backbones and reasoning datasets. For instance, on Qwen3-0.6B, it surpasses GPTQ by 44.53% on MATH-500 and consistently recovers performance in the 2-bit regime.


## 1 Introduction

Recent large language models (LLMs) (Jaech et al., 2024; Guo et al., 2025; Team et al., 2025) with enhanced reasoning capabilities have achieved remarkable progress in domains such as coding and mathematics. However, this progress comes with a deployment bottleneck: reasoning-focused inference is often slow and token-inefficient, resulting in substantial inference overhead (Qu et al., 2025). Quantization is a widely used technique to accelerate LLM inference (Frantar et al., 2022a; Lin et al., 2023; Li et al., 2024; Liu et al., 2024; Li et al., 2025a; Ma et al., 2024; Lin et al., 2024), yet prior studies (Li et al., 2025b; Srivastava et al., 2025; Liu et al., 2025b; Wang et al., 2025) show that under extreme low-bit settings (e.g., 3-bit or 2-bit weight-only quantization), post-training quantization (PTQ) can trigger severe accuracy degradation on reasoning benchmarks. We corroborate this phenomenon by comparing quantized LLMs on both non-reasoning and reasoning tasks (Figure 1): while 4-bit group-wise weight quantization (group size 128) is near-lossless across tasks, 3-bit variants incur large drops, and the degradation is remarkably larger on reasoning tasks than on non-reasoning ones.

Quantization-aware training (QAT) (Tailor et al., 2020; Nagel et al., 2022; Bondarenko et al., 2024; Jeon et al., 2024; Qin et al., 2024) is an appealing alternative to recover the performance drop by simulating low-precision inference during training. While QAT has demonstrated effectiveness for general-purpose LLMs (Liu et al., 2023; Chen et al., 2024), it remains unclear whether these benefits extend to reasoning-focused models, and it is non-trivial to address. For instance, our preliminary attempts show that applying reinforcement learning (RL) directly on a severely degraded quantized policy often fails to explore valid reasoning trajectories under quantization noise. This motivates this study: *what are the key factors that lead to the success of QAT with reasoning models?*

In this study, we present a systematic study of quantization-aware training (QAT) for reasoning models. We investigate the following critical factors: 1) the choice of training objective, e.g., supervised fine-tuning (SFT) vs. knowledge distillation (KD) (Hinton et al., 2015); 2) the role of PTQ initialization; 3) the integration of RL with QAT; and 4) the choice of QAT training data. We study two representative quantization settings—3-bit and 2-bit weight-only quantization with group size 128, covering two major reasoning training paradigms: (i) supervised fine-tuning (SFT), represented by DeepSeek-R1-Qwen-Distill-1.5B (Guo et al., 2025), and (ii) reinforcement learning (RL), represented by Qwen3-0.6B and Qwen3-4B (Yang et al., 2025). We evaluate on a diverse suite of reasoning benchmarks, including AIME-120, MATH-500 (Lightman et al., 2023), GSM8K (Cobbe et al., 2021), GPQA-Diamond (Rein et al., 2024), and LiveCodeBench (Jain et al., 2024). We summarize our findings as follows:

- **Training Objective** (§3.2): Knowledge distillation (KD) (Hinton et al., 2015) is the preferred objective for QAT, which can effectively boost reasoning models trained by either SFT or RL.
- **PTQ Initialization** (§3.3): PTQ provides a strong initialization that effectively saves the training cost and stabilizes the QAT training, especially in early stages.
- **QAT with Reinforcement Learning** (§3.4): With KD training as the cold start, QAT with RL can deliver further performance gains.
- **Choice of QAT Data** (§3.5): Aligning the domain of QAT dataset with the calibration set by PTQ is beneficial for QAT training, i.e., it yields a faster and more stable training process.

Finally, based on the findings above, we optimize the QAT workflow for reasoning models, termed as *Reasoning-QAT*. Specifically, Reasoning-QAT is structured in the following way: *PTQ-based initialization* $\rightarrow$ *KD-based recovery* $\rightarrow$ *Cold-start RL*. Our empirical results show that this workflow consistently outperforms state-of-the-art PTQ and QAT baselines across multiple backbones and reasoning benchmarks. For example, on Qwen3-0.6B, it surpasses GPTQ (Frantar et al., 2022a) by 44.53% on MATH-500 under 3-bit quantization; meanwhile, on DeepSeek-R1-Distill-Qwen-1.5B, it outperforms representative QAT baselines by up to 4.75% on average. We hope our research provides valuable guidance toward better quantization methods for reasoning models.

## 2 Preliminaries and Research Questions

### 2.1 Post-training Quantization for Reasoning Models

**Background and Notations.** Quantization has been a popular approach for the compression and acceleration of LLMs. Given model parameters $\mathbf{W}$ stored in bfloat16, quantization converts $\mathbf{W}$ into low-bit integer representations $\mathbf{W}_{int}$, i.e.,

$$\mathbf{W}_{int} = \text{clip}(\lfloor \frac{\mathbf{W}}{s} \rceil + z, Q_{min}, Q_{max}), \quad (1)$$

where $\text{clip}(\cdot,\ Q_{min},\ Q_{max})$ clips values into the range $[Q_{min}, Q_{max}]$, $s$ is the scaling factor and $z$ is the zero point. For $N$-bit symmetric quantization, $s = \frac{\max(|\mathbf{W}|)}{2^{N-1}-1}$ and $z = 0$. For asymmetric quantization, $s = \frac{max(\mathbf{W})-min(\mathbf{W})}{2^N-1}$, $z = \lfloor \frac{-min(\mathbf{W})}{s} \rceil$.

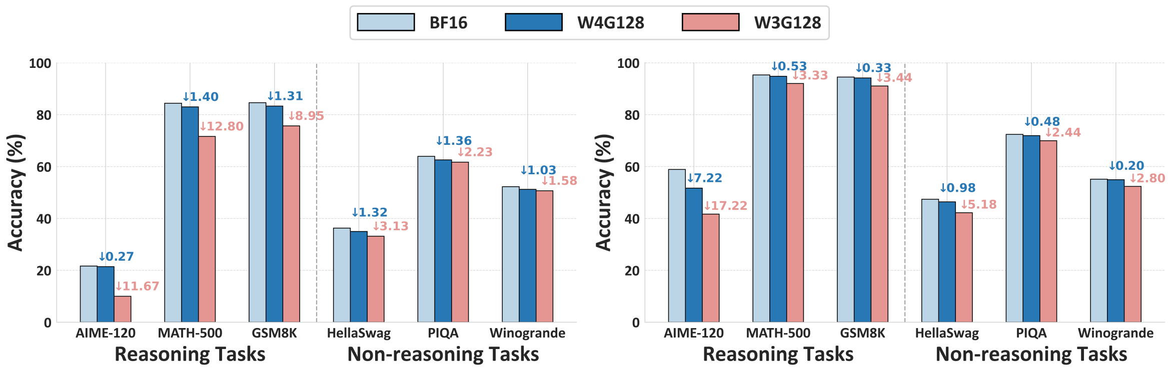
For weight quantization, the low bit quantized weights $\mathbf{W}_{int}$ in the forward pass are then dequantized to $\hat{\mathbf{W}} = s \cdot (\mathbf{W_{int}} - z)$ for the following operations. For completeness, in weight-activation quantization, both weights and activations are stored as low-bit integers and computed with integer kernels, which can further reduce compute beyond memory savings.

**Post-training quantization incurs a large performance drop on reasoning models.** Most prior work on LLM quantization focuses on post-training quantization (PTQ) (Frantar et al., 2022b; Lin et al., 2023; Ashkboos et al., 2024; Sun et al., 2024; Liu et al., 2025a), where the model is directly quantized without training. PTQ is usually fast and easy to implement, with satisfactory performance on many general natural language tasks. However, recent studies (Liu et al., 2025b) show that quantized reasoning models can exhibit large performance drops, particularly on challenging reasoning benchmarks.

To further validate this, we compare PTQ-quantized LLMs on both non-reasoning and reasoning benchmarks. From Figure 1, it can be found that for DeepSeek-R1-Distill-Qwen-1.5B (abbr. R1-Qwen-1.5B for simplicity in the following text), the performance degradation on reasoning tasks (e.g., 11.67%↓ on AIME-120 and 12.80%↓ on MATH-500) is much larger than that on non-reasoning tasks (e.g., 1.03%↓ on Winogrande, 3.13%↓ on Hellaswag). Similar observations can be found for Qwen3-4B. Therefore, under extreme low-bit settings, PTQ alone is often insufficient to preserve reasoning performance, motivating training-time adaptation such as QAT.

### 2.2 Quantization-aware Training for Reasoning Models

To mitigate the performance degradation of PTQ, QAT is a commonly used alternative. QAT simulates low-precision inference during training, allowing model weights to adapt to quantization effects.




(a) DeepSeek-R1-Distill-Qwen-1.5B. (b) Qwen3-4B.

Figure 1: The performance degradation by post-training quantization on reasoning and non-reasoning tasks. We adopt GPTQ with 3-bit weight only quantization with group size 128, and the results are based on DeepSeek-R1-Distill-Qwen-1.5B and Qwen-4B.

In the forward pass, QAT inserts fake quantization operations to obtain quantized weights $\hat{\mathbf{W}}$ or activations $\hat{\mathbf{X}}$ in each linear layer, and optimizes the training objective $\mathcal{L}(\hat{\mathbf{W}})$. In the backward pass, since quantization is non-differentiable, the straight-through estimator (STE) is typically used to pass gradients to the original weights $\mathbf{W}$, e.g., $\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{W}}} \cdot \mathbf{1}(Q_{min} \leq \mathbf{W}/s \leq Q_{max})$.

While QAT has been shown effective for general-purpose LLMs (Liu et al., 2023; Chen et al., 2024), how these benefits extend to reasoning models remains unclear. In this study, we aim to investigate the following four research questions (RQs):

**RQ1.** Which training objective is most suitable for QAT on reasoning models?

**RQ2.** What improves the training efficiency of QAT under low-bit settings?

**RQ3.** How does QAT interact with RL (e.g., GRPO) in the low-bit regime?

**RQ4.** How does the choice of QAT training data affect quantized reasoning performance?

**Training Paradigm and Objectives (RQ1 & RQ3).** The optimal training methodology for QAT on reasoning models is unclear. Standard QAT often reuses the cross-entropy objective from pre-training or instruction fine-tuning (Liu et al., 2025c; Lee et al., 2024). In contrast, many reasoning models are trained either by supervised fine-tuning with teacher-generated data (often paired with knowledge distillation) (Guo et al., 2025) or by reinforcement learning (Guo et al., 2025; Team et al., 2025; Yang et al., 2025). How to integrate the training objectives (e.g., SFT vs. KD) as prerequisites for stable low-bit training, and the combination of QAT with RL, remain unexplored.

**Training Efficiency and Overhead (RQ2).** Severe accuracy degradation under extreme low-bit quantization often requires substantial additional training to recover reasoning performance. This can make QAT time- and compute-intensive, hindering practical deployment when the retraining budget is limited. Identifying strategies to improve sample and time efficiency of QAT is therefore a major practical concern.

**Data Strategy (RQ4).** Beyond the training objective, the selection of data for QAT itself is a critical factor. The impact of QAT training data (including its domain, quality, and alignment with calibration data) on the reasoning performance of the quantized model is not well characterized. A systematic analysis is needed to guide efficient and effective data curation.

## 3 A Systematic Study of QAT for Reasoning Models

In this study, we conduct a systematic empirical study of quantization-aware training for reasoning models, aiming to address the research questions posed in §2.2.

### 3.1 Setups

**Quantization Settings.** We quantize all linear layers of the model, excluding the token embedding and `lm_head` layers. Our primary focus is on 3-bit and 2-bit group-wise weight-only quantization with a group size of 128 (W3G128 and W2G128). For initialization, we consider two commonly used schemes: a symmetric round-to-nearest (RTN) baseline and an asymmetric GPTQ initialization. For completeness, we also report results under a joint 4-bit weight and 4-bit activation (W4A4) setting in Appendix A.3.

**Models and Dataset.** We evaluate two categories of reasoning models. For SFT-based reasoning models, we adopt DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025). For RL-trained reasoning models, we use Qwen3-0.6B and Qwen3-4B (Yang et al., 2025), the two recent competitive open-source LLMs. For the choice of training dataset, our study includes two training phases with distinct data configurations. In the initial fine-tuning phase (SFT and KD), we use OpenR1-Math (Face, 2025) (94k problems in the default subset) for the weight-only low-bit setting. In the subsequent reinforcement learning (RL) phase, we also use the OpenR1-Math dataset. This design allows us to examine the effect of domain consistency between the QAT training data and the calibration data (§3.5). For completeness, the data configuration for the W4A4 setting is reported in Appendix A.3.

**Evaluation Benchmarks.** We assess quantized models across training paradigms on a suite of reasoning benchmarks: (1) three mathematical reasoning benchmarks sorted by difficulty: AIME-120 (120 problems from AIME 2022–2025), MATH-500 (Lightman et al., 2023), and GSM8K (Cobbe et al., 2021); (2) LiveCodeBench (Jain et al., 2024) for code generation; and (3) GPQA-Diamond (Rein et al., 2024) for graduate-level science question answering. All evaluations are conducted with LightEval (Fourrier et al., 2023) using the vLLM (Kwon et al., 2023) backend. We set temperature to 0.6, top-$p$ to 0.95, and the maximum number of generated tokens to 32,768. We report average scores over three random seeds.

**Training Implementations.** We implement and evaluate three training objectives in our study. For supervised fine-tuning (SFT), we use the standard cross-entropy loss. For knowledge distillation (KD), the quantized model serves as the student and is trained to match the output distribution of the full-precision teacher by minimizing KL divergence. For reinforcement learning (RL), we employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Further hyperparameter details are provided in Appendix A.1.

| Model | Setting | Method | AIME120 | MATH-500 | GSM8K | AVG | Drop↓ |
|---|---|---|---|---|---|---|---|
| R1-Qwen-1.5B | BF16 | - | 21.67 | 84.4 | 84.61 | 63.56 | – |
| | W3G128 | RTN | 0.83 | 15.00 | 15.39 | 10.41 | 53.15↓ |
| | | SFT | 10.00 | 73.60 | 75.54 | 53.05 | 10.51↓ |
| | | KD | 14.44 | 76.20 | 75.87 | 55.50 | 8.06↓ |
| Qwen3-4B | BF16 | - | 58.89 | 95.33 | 94.49 | 82.90 | – |
| | W3G128 | RTN | 0.00 | 1.40 | 0.99 | 0.80 | 82.10↓ |
| | | SFT | 14.44 | 81.80 | 88.25 | 61.50 | 21.40↓ |
| | | KD | 37.50 | 92.00 | 91.43 | 73.64 | 9.26↓ |

Table 1: Objective choice for low-bit QAT. KD yields smaller accuracy drops than SFT on both an SFT-trained model (R1-Qwen-1.5B) and an RL-trained model (Qwen3-4B) under W3G128, while SFT degrades much more on the RL-trained model.

### 3.2 QAT Training Objectives: SFT or KD?

We investigate QAT objectives (SFT vs. KD) for reasoning models trained via supervised fine-tuning (e.g., R1-Qwen-1.5B) or reinforcement learning (e.g., Qwen3-4B). Adopting W3G128 weight-only quantization, Table 1 compares the performance recovery of KD and SFT. Key observations include: 1) KD outperforms SFT on both model types. Specifically, SFT suffers average accuracy drops of 10.51%↓ and 21.40%↓ on R1-Qwen-1.5B and Qwen3-4B, respectively, whereas KD limits drops to 8.06%↓ and 9.26%↓; 2) KD exhibits consistent synergy across paradigms, with similar drops for both models (8.06% vs. 9.26%). In contrast, SFT causes a moderate drop on R1-Qwen-1.5B but a more severe one (21.40%↓) on Qwen3-4B. We hypothesize that KD provides smoother signals by aligning output distributions, preserving uncertainty structure better than hard-label SFT. *We therefore recommend KD over SFT for QAT due to its superior performance and robustness across training paradigms.*

**Findings (RQ1)**

- KD is preferable to SFT as the QAT objective for reasoning models, and is more consistent across both SFT-trained and RL-trained reasoning models.

### 3.3 Training Efficiency of QAT

**Initializing QAT with PTQ.** In previous work, it has been standard practice to initialize QAT from a pretrained full-precision model (Liu et al., 2023; Du et al., 2024). Here, we systematically investigate how PTQ-based initializations affect the convergence and accuracy of QAT. Specifically, we employ GPTQ (Frantar et al., 2022a) for initialization, using weights that have been adjusted via Hessian-based compensation prior to the quantization process. As shown in Figure 2(a)-(b), we compare the test accuracy and training loss of RTN+KD and GPTQ+KD on the MATH-500 benchmark using the R1-Qwen-1.5B model. Using the GPTQ-initialized weights, GPTQ+KD starts from a higher starting point (higher test accuracy and lower loss). Furthermore, GPTQ+KD consistently outperforms RTN+KD and exhibits a faster convergence rate within the same number of training steps. This is likely because PTQ provides a more accurate starting point than RTN, which minimizes the initial performance gap and mitigates optimization difficulties. Therefore, *PTQ acts as a strong initialization to improve the training efficiency of QAT.*

**Training Efficiency: KD vs. SFT.** In addition to studying the initialization of quantized models, we further compare the training efficiency of KD versus SFT, as shown in Figure 3. *The results show that KD consistently achieves higher accuracy than SFT and also converges faster.*

**Findings (RQ2)**

- PTQ acts as a strong initialization to minimize the initial performance gap, while KD converges faster than SFT, collectively improving training efficiency and accuracy.

### 3.4 QAT with Reinforcement Learning

While Reinforcement Learning (RL) enhances LLM reasoning, its role in QAT remains underexplored. Our results identify a critical prerequisite: *RL requires proper initialization to avoid collapse*. We compare *zero-RL QAT* (applying RL directly to RTN-quantized models) and *cold-start RL QAT* (initializing with a KD-tuned model). Using GRPO (Guo et al., 2025) with correctness rewards, Table 2 shows that zero-RL QAT collapses completely, whereas the cold-start setting improves accuracy by approximately 46%. RTN-based models with severe quantization errors fail to sample valid outputs, preventing effective reward generation. Conversely, KD recovers sampling capability, ensuring sufficient reward density. We conclude: *RL alone cannot rescue heavily quantized models but drives improvement given a KD cold start.*

**Roles of RL under low-bit QAT.** Figure 4 demonstrates the indispensable roles of RL in QAT.

| RTN | KD | GRPO | AIME120 | MATH-500 | GSM8K | AVG |
|---|---|---|---|---|---|---|
| - | - | - | 21.67 | 84.40 | 84.61 | 63.56 |
| ✓ | - | - | 0.83 | 15.00 | 15.39 | 10.41 |
| ✓ | - | ✓ | 1.67 | 15.33 | 15.52 | 10.84 |
| ✓ | ✓ | ✓ | 14.44 | 78.00 | 77.93 | 56.79 |

Table 2: RL under low-bit quantization (R1-Qwen-1.5B, W3G128). KD cold-start is necessary for effective RL.

First, as shown in Figure 4(a), RL simultaneously increases reward and reduces excessive response length. This prevents the model from exploiting response length to increase rewards, guiding it toward high-quality outputs. Second, in Figure 4(b), RL drives a decrease in entropy, which reduces prediction randomness and enforces deterministic outputs. This avoids collapse and ensures stable convergence despite quantization errors. Lastly, in Figure 4(c), RL improves test accuracy while reducing response length, enhancing generalization without unnecessary verbosity. We hypothesize that RL works by reducing entropy, helping the model resist quantization noise.

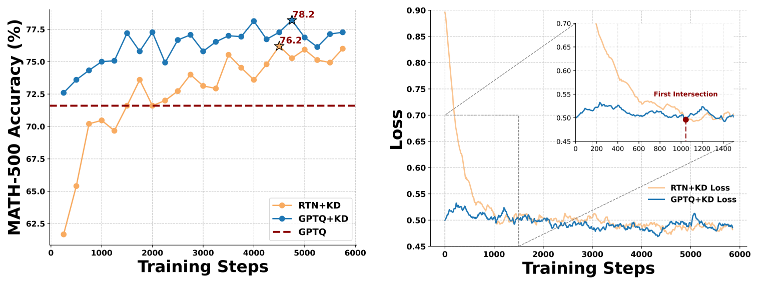
**Findings (RQ3)**

- KD cold-start serves as a practical prerequisite for RL under low-bit QAT, enabling subsequent RL to enhance both model performance and efficiency.

### 3.5 The Choice of QAT Training Dataset

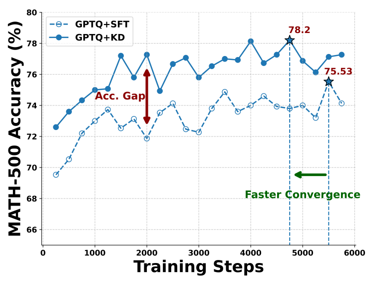
The choice of the QAT training dataset remains an open challenge, specifically regarding how domain differences influence optimization dynamics and final performance. We compare two datasets: Wikitext2 (natural language) and OpenR1-Math (reasoning-based math). Following §3.3, we initialize QAT models using PTQ. Calibration uses either Wikitext2 or NuminaMath-1.5, where the latter is closely aligned with OpenR1-Math[1]. We then conduct KD for QAT on both datasets, yielding four combinations. Figure 5 shows test accuracy curves, revealing three observations: First, domain alignment accelerates convergence: when the calibration and the training domains match, the curve converges much earlier (e.g., `numina+openr1`), whereas mismatched combinations stabilize slowly, changing beyond 5k+ steps. Second, domain mismatch causes harmful readjustment. Specifically, `numina+wiki` starts

[1]OpenR1-Math consists of reasoning traces generated by DeepSeek R1 for problems from NuminaMath 1.5. See `https://huggingface.co/datasets/open-r1/OpenR1-Math-220k`.



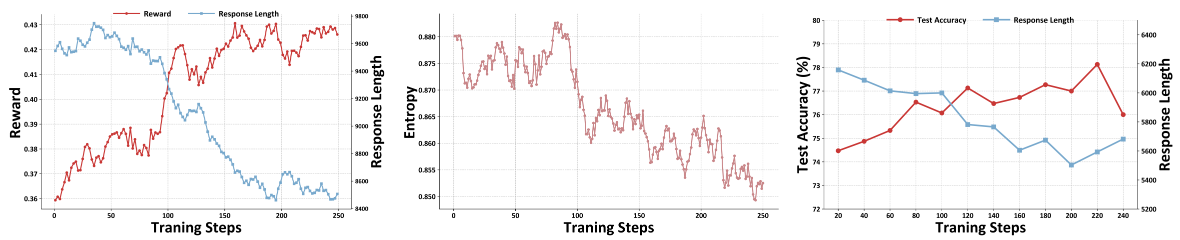

(a) Test Accuracy (b) Loss Value

Figure 2: PTQ initialization improves QAT efficiency (RTN+KD vs. GPTQ+KD on MATH-500): (a) accuracy, (b) loss.




(a) Test Accuracy

Figure 3: KD vs. SFT under GPTQ initialization on MATH-500.




(a) (b) (c)

Figure 4: RL effects after KD cold-start (W3G128). (a) reward↑ with length↓, (b) entropy↓, (c) test accuracy↑ with length↓.

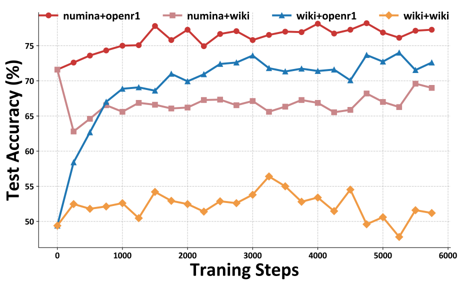



Figure 5: Comparison with different combinations of PTQ and QAT data domain. We present test accuracy curves on MATH-500 for the R1-Qwen-1.5B model under W3G128 setting.

with strong initialization but suffers a sharp accuracy drop after switching to Wikitext2, recovering slowly without returning to the initial level. This implies the training domain significantly alters the PTQ-calibrated solution, making cross-domain adaptation unstable. Third, reasoning-domain data is critical for final performance: configurations using reasoning data (especially `numina+openr1`) achieve the highest accuracy, while general text (`wiki+wiki`) leads to substantially lower accuracy. Overall, these results suggest aligning PTQ calibration and QAT training domains, and confirm that reasoning data is crucial for recovering low-bit performance.

**Findings (RQ4)**

- Aligning the PTQ calibration domain with the QAT training domain accelerates convergence, whereas utilizing reasoning-domain data is key to achieving a high final reasoning performance.

# 4 Reasoning-QAT: The Optimized Workflow

Based on the observations in §3, we provide the optimized workflow in Figure 6, which includes three key steps to guide practical applications and support downstream usage. Note that this section does not propose a new algorithm; rather, it validates our findings and offers feasible guidance, showing that satisfying the identified dependencies is sufficient for stable low-bit training. For completeness, we include an additional weight-activation setting in Appendix A.3.

## 4.1 The Reasoning-QAT Workflow

We construct this reference workflow by strictly satisfying the dependencies identified in RQ1–RQ3:

- **PTQ-based Initialization.** Motivated by §3.3, we rectify the latent weights with PTQ techniques as the initial state for QAT. While the

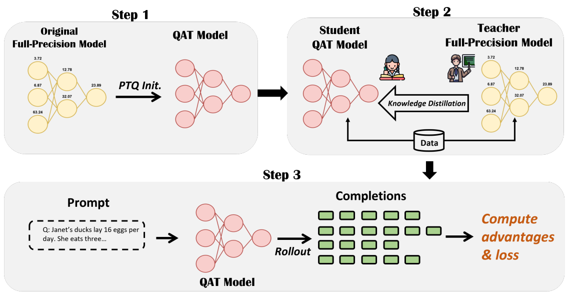



Figure 6: The overall workflow of the Reasoning-QAT. Step 1: PTQ-based initialization can provide a better starting point. Step 2: KD from the original full-precision model to align the teacher's behavior, and also serve as a cold-start for subsequent RL. Step 3: Based on cold-start, RL can further recover the reasoning ability of the QAT model.

QAT model still retains continuous weights, this initialization strategy improves its tolerance to quantization and provides a better starting point for subsequent training.

- **Knowledge Distillation.** We then perform knowledge distillation from the original full-precision model. Guided by the findings in §3.2, this step fine-tunes the QAT model to align its output distribution with that of the full-precision model. After that, the distilled model not only recovers from the quantization-induced degradation, but also serves as a stable cold-start actor for RL.
- **Cold-start RL.** Following the prerequisites discussed in §3.4, we apply RL on top of the knowledge-distilled model from Step 2. Here, we employ GRPO (Guo et al., 2025) as the RL paradigm. This cold start design avoids the collapse issue observed when directly using RL on heavily quantized models, while utilizing the stabilized initialization to ensure reliable optimization. During this stage, RL progressively enhances the reasoning capability of the quantized model, driving more deterministic outputs and reducing randomness.

## 4.2 Empirical Evaluations

**Comparison with Representative PTQ Baselines.** We compare Reasoning-QAT against commonly used PTQ baselines under two settings: 3-bit weight-only (W3G128) and 2-bit weight-only (W2G128) quantization.

**3-Bit Weight-only Quantization.** Table 3 shows that PTQ baselines (e.g., RTN/GPTQ/AWQ) can incur severe degradation on reasoning benchmarks in the 3-bit regime. In contrast, Reasoning-QAT consistently recovers a substantial portion of the lost accuracy across model scales. For example, on Qwen3-0.6B the average score increases from 12.61% (GPTQ) to 31.67% (Reasoning-QAT). On R1-Qwen-1.5B and Qwen3-4B, the remaining gap to BF16 is reduced to -5.46% and -8.25%, respectively, outperforming the PTQ baselines under the same setting. Overall, these results suggest that when PTQ alone struggles at low bits, the dependency-satisfying workflow provides a strong empirical recovery baseline.

**2-Bit Weight-only Quantization.** The 2-bit setting (W2G128) is substantially more challenging, and PTQ baselines largely collapse on math-centric benchmarks. In contrast, Reasoning-QAT recovers non-trivial reasoning accuracy, with particularly large gains on math datasets. For instance, on R1-Qwen-1.5B, MATH-500 improves from 3.67% (GPTQ) to 55.00%. On Qwen3-4B, MATH-500 improves from 4.80% (GPTQ) to 78.27% and AIME-120 reaches 22.78%. This highlights the necessity of QAT under extremely low-bit quantization.

**Comparison with Representative QAT Baselines.** We additionally compare Reasoning-QAT with two representative QAT methods developed for general-purpose LLMs, EfficientQAT (Chen et al., 2024) and BitDistiller (Du et al., 2024). Because these methods were not originally reported on reasoning benchmarks, we reproduce them under the same protocol (R1-Qwen-1.5B, W3G128, OpenR1-Math) for a fair comparison.

Table 4 shows that Reasoning-QAT achieves the best average accuracy (58.51%) among the reproduced QAT baselines. The gap is moderate but consistent (e.g., +4.75% over EfficientQAT and +1.45% over BitDistiller), supporting our central observation that low-bit QAT for reasoning models: KD provides a stronger recovery objective, and once a viable policy is established, RL can further improve model performance.

## 4.3 Analysis of Workflow Components

In this section, we clarify the efficacy of each workflow component, which are PTQ initialization, KD, and GRPO. We specifically assess W3G128 on R1-Qwen-1.5B model shown in Table 8 (detailed analysis in Appendix A.2).

| Model | W-Bits | Methods | AIME-120 | MATH-500 | GSM8K | GPQA-Diamond | LiveCode-Bench | Avg. | Drop ↓ |
|---|---|---|---|---|---|---|---|---|---|
| Qwen3-0.6B | BF16 | - | 11.11 | 74.00 | 79.00 | 28.45 | 12.94 | 41.10 | - |
| | W3G128 | RTN | 0.00 | 0.80 | 0.30 | 24.24 | 0.00 | 5.07 | -36.03 |
| | | GPTQ | 0.00 | 13.27 | 23.33 | 26.43 | 0.00 | 12.61 | -28.49 |
| | | AWQ | 0.00 | 5.20 | 10.01 | 26.77 | 0.00 | 8.40 | -32.70 |
| | | Reasoning-QAT | 3.89 | 57.80 | 67.02 | 27.78 | 1.87 | 31.67 | -9.43 |
| | W2G128 | GPTQ | 0.00 | 0.60 | 0.13 | 0.84 | 0.00 | 0.31 | -40.79 |
| | | AWQ | 0.00 | 0.00 | 0.00 | 23.91 | 0.00 | 4.78 | -36.32 |
| | | Reasoning-QAT | 0.56 | 20.20 | 30.33 | 25.25 | 0.00 | 15.27 | -25.83 |
| R1-Qwen-1.5B | BF16 | - | 21.67 | 84.40 | 84.61 | 36.87 | 16.04 | 48.72 | - |
| | W3G128 | RTN | 0.83 | 15.00 | 15.39 | 19.19 | 0.00 | 10.08 | -38.64 |
| | | GPTQ | 10.00 | 71.60 | 75.66 | 23.74 | 9.33 | 38.07 | -10.65 |
| | | AWQ | 3.33 | 48.80 | 65.81 | 37.88 | 4.85 | 32.13 | -16.58 |
| | | Reasoning-QAT | 16.39 | 79.80 | 79.35 | 30.30 | 10.45 | 43.26 | -5.46 |
| | W2G128 | GPTQ | 0.00 | 3.67 | 2.86 | 21.89 | 0.00 | 5.68 | -43.04 |
| | | AWQ | 0.00 | 0.00 | 0.00 | 25.08 | 0.00 | 5.02 | -43.70 |
| | | Reasoning-QAT | 5.15 | 55.00 | 55.02 | 25.75 | 0.00 | 28.18 | -20.54 |
| Qwen3-4B | BF16 | - | 58.89 | 95.33 | 94.49 | 56.06 | 48.38 | 70.63 | - |
| | W3G128 | RTN | 0.00 | 1.40 | 0.99 | 10.60 | 0.00 | 2.60 | -68.03 |
| | | GPTQ | 34.17 | 90.07 | 91.74 | 38.05 | 20.77 | 54.96 | -15.67 |
| | | AWQ | 25.00 | 87.00 | 90.07 | 37.88 | 19.03 | 51.80 | -18.83 |
| | | Reasoning-QAT | 41.11 | 93.47 | 93.48 | 45.79 | 38.06 | 62.38 | -8.25 |
| | W2G128 | GPTQ | 0.00 | 4.80 | 5.28 | 20.70 | 0.00 | 6.16 | -64.47 |
| | | AWQ | 0.00 | 0.00 | 0.00 | 25.59 | 0.00 | 5.12 | -65.61 |
| | | Reasoning-QAT | 22.78 | 78.27 | 82.96 | 25.42 | 0.00 | 41.89 | -28.74 |

Table 3: Main results of the reference configuration (Reasoning-QAT) versus representative PTQ baselines across models and reasoning benchmarks.

| Method | AIME120 | MATH-500 | GSM8K | AVG |
|---|---|---|---|---|
| Standard SFT-QAT | 10.00 | 73.60 | 75.54 | 53.05 |
| EfficientQAT (Chen et al., 2024) | 10.83 | 74.20 | 76.26 | 53.76 |
| BitDistiller (Du et al., 2024) | 14.72 | 78.00 | 78.46 | 57.06 |
| **Reasoning-QAT** | **16.39** | **79.80** | **79.35** | **58.51** |

Table 4: Comparison with representative QAT baselines on R1-Qwen-1.5B (W3G128) under the same protocol (OpenR1-Math). EfficientQAT and BitDistiller are reproduced for this setting. The best result is shown in bold.

| Configuration | AIME120 | MATH-500 | GSM8K | AVG |
|---|---|---|---|---|
| BF16 | 21.67 | 84.40 | 84.61 | 63.56 |
| RTN | 0.83 | 15.00 | 15.39 | 10.41 |
| GPTQ (PTQ init) | 10.00 | 71.60 | 75.66 | 52.42 |
| GPTQ + SFT | 14.17 | 75.53 | 76.12 | 55.27 |
| GPTQ + KD | 13.89 | 78.20 | 77.26 | 56.45 |
| **GPTQ + KD + GRPO** | **16.39** | **79.80** | **79.35** | **58.51** |

Table 5: Ablation study on R1-Qwen-1.5B (W3G128), full results are reported in Appendix A.2 (Table 8). The best result in each case is shown in bold.

**The Effect of PTQ Initialization.** GPTQ yields a markedly better low-bit starting point than RTN (AVG: 52.42% vs. 10.41%), which is also consistent with the findings in §3.3.

**The Effect of KD.** Under the same GPTQ initialization, KD improves over SFT from 55.27% to 56.45%, indicating that distillation remains beneficial even when starting from a strong PTQ baseline. The result aligns with §3.2 that KD is the preferred objective for low-bit QAT on reasoning models.

**Further Improvement by GRPO.** Finally, adding GRPO on top of KD yields additional gains: 56.45% $\rightarrow$ 58.51% under GPTQ. This is consistent with §3.4: RL is effective after KD cold start has established a viable low-bit policy, at which point RL can further improve performance. Overall, the workflow (GPTQ+KD+GRPO) reaches 58.51% on AVG, which narrows the gap to the BF16 upper bound, i.e., 63.56%.

## 5 Conclusion

We present a systematic empirical study of QAT for reasoning models under extreme low-bit quantization. We identify four key findings: KD is the preferred recovery objective, PTQ initialization improves training efficiency and accuracy, RL provides additional gains when a viable cold start exists (with KD acting as a practical prerequisite), and PTQ–QAT domain alignment affects convergence and accuracy. Finally, we consolidate these findings into Reasoning-QAT, a three-stage workflow that validates our findings and achieves a strong recovery compared to PTQ baselines.

## Limitations

First, our QAT training utilizes primarily math-centric data. While this effectively recovers performance on mathematical benchmarks, cross-domain generalization to other reasoning tasks (e.g., coding and science) remains limited, particularly for smaller models under extreme low-bit settings. Second, as an empirical study, this work focuses on validating optimal training workflows rather than designing new quantization methods.

## References


Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. arXiv preprint arXiv:2404.00456.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 7432–7439.

Yelysei Bondarenko, Riccardo Del Chiaro, and Markus Nagel. 2024. Low-rank quantization-aware training for llms. arXiv preprint arXiv:2406.06385.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. 2024. Efficientqat: Efficient quantization-aware training for large language models. arXiv preprint arXiv:2407.11062.

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. arXiv preprint arXiv:2110.14168.

Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. 2024. Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation. arXiv preprint arXiv:2402.10631.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Clémentine Fourrier, Nathan Habib, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. Lighteval: A lightweight framework for llm evaluation.

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

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022b. Optq: Accurate quantization for generative pre-trained transformers. In The Eleventh International Conference on Learning Representations.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. arXiv preprint arXiv:2412.16720.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. arXiv preprint arXiv:2403.07974.

Hyesung Jeon, Yulhwa Kim, and Jae-joon Kim. 2024. L4q: Parameter efficient quantization-aware training on large language models via lora-wise lsq. CoRR.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the 29th Symposium on Operating Systems Principles, pages 611–626.

Janghwan Lee, Seongmin Park, Sukjin Hong, Minsoo Kim, Du-Seong Chang, and Jungwook Choi. 2024. Improving conversational abilities of quantized large language models via direct preference alignment. arXiv preprint arXiv:2407.03051.

Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. 2024. Evaluating quantized large language models. arXiv preprint arXiv:2402.18158.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. 2025a. Gptaq: Efficient finetuning-free quantization for asymmetric calibration. arXiv preprint arXiv:2504.02692.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl

Cobbe. 2023. Let's verify step by step. In The Twelfth International Conference on Learning Representations.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. 2024. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. Advances in Neural Information Processing Systems, 37:87766–87800.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration. arXiv preprint arXiv:2306.00978.

Fangxin Liu, Zongwu Wang, JinHong Xia, Junping Zhao, Jian Liu, Haibing Guan, and Li Jiang. 2025a. Flexquant: A flexible and efficient dynamic precision switching framework for llm quantization. arXiv preprint arXiv:2506.12024.

Ruikang Liu, Haoli Bai, Haokun Lin, Yuening Li, Han Gao, Zhengzhuo Xu, Lu Hou, Jun Yao, and Chun Yuan. 2024. Intactkv: Improving large language model quantization by keeping pivot tokens intact. arXiv preprint arXiv:2403.01241.

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

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023. Llm-qat: Data-free quantization aware training for large language models. arXiv preprint arXiv:2305.17888.

Zechun Liu, Changsheng Zhao, Hanxian Huang, Sijia Chen, Jing Zhang, Jiawei Zhao, Scott Roy, Lisa Jin, Yunyang Xiong, Yangyang Shi, and 1 others. 2025c. Paretoq: Scaling laws in extremely low-bit llm quantization. arXiv preprint arXiv:2502.02631.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. 2024. Affinequant: Affine transformation quantization for large language models. arXiv preprint arXiv:2403.12544.

Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. 2022. Overcoming oscillations in quantization-aware training. In International Conference on Machine Learning, pages 16318–16330. PMLR.

Haotong Qin, Xudong Ma, Xingyu Zheng, Xiaoyang Li, Yang Zhang, Shouda Liu, Jie Luo, Xianglong Liu, and Michele Magno. 2024. Accurate lora-finetuning quantization of llms via information retention. arXiv preprint arXiv:2402.05445.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, and 1 others. 2025. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. arXiv preprint arXiv:2503.21614.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In First Conference on Language Modeling.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

Gaurav Srivastava, Shuxiang Cao, and Xuan Wang. 2025. Towards reasoning ability of small language models. arXiv preprint arXiv:2502.11569.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, and 1 others. 2024. Flatquant: Flatness matters for llm quantization. arXiv preprint arXiv:2410.09426.

Shyam A Tailor, Javier Fernandez-Marques, and Nicholas D Lane. 2020. Degree-quant: Quantization-aware training for graph neural networks. arXiv preprint arXiv:2008.05000.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. arXiv preprint arXiv:2507.20534.

Hongyu Wang, Shuming Ma, and Furu Wei. 2025. Bitnet v2: Native 4-bit activations with hadamard transformation for 1-bit llms. arXiv preprint arXiv:2504.18415.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. arXiv preprint arXiv:2505.09388.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830.

# A Appendix

## A.1 Training Implementations Details

We list the detailed training hyperparameters in Tables 6 and 7.

| Parameter | W3G128 | | | | W2G128 |
|---|---|---|---|---|---|
| | RTN Init | | GPTQ Init | | GPTQ Init. |
| | SFT | KD | SFT | KD | KD |
| ***Optimizer*** | | | | | |
| Optimizer | Adam | | | | |
| LR (Peak) | 2e-5 | 2e-5 | 1e-6 | 1e-6 | 5e-5 |
| LR Scheduler | Cosine Decay | | | | |
| Warmup Steps | 180 | 180 | 180 | 180 | 180 |
| Adam Betas ($\beta_1$, $\beta_2$) | 0.9, 0.95 | | | | |
| ***Training*** | | | | | |
| Global Batch Size | 32 | | | | |
| Gradient Accumulation | 4 | | | | |
| Training Steps | 6,000 | 6,000 | 6,000 | 6,000 | 6,000 |

Table 6: Hyperparameters for Phase 1 (Cold Start).

| Parameter | Value |
|---|---|
| ***Optimizer*** | |
| Optimizer | Adam |
| LR (Peak) | 5e-7 |
| Scheduler | Cosine |
| Warmup | 8 |
| Betas | 0.9, 0.95 |
| ***Training*** | |
| Batch Size | 64 |
| Grad Acc | 4 |
| Steps | 250 |
| ***Rollout*** | |
| Group Size | 8 |
| Max Len | 32768 |

Table 7: Hyperparameters for Phase 2 (RL).

## A.2 Complete Factorial Combinations on Weight-Only Settings

In this section, we clarify the efficacy of each Reasoning-QAT components, which are PTQ initialization, KD, and GRPO. We specifically assess the 3-bit groupwise weight-only quantization on R1-Qwen-1.5B model shown in Table 8.

**The Effect of PTQ Initialization.** To investigate the impact of different weight quantization initialization strategies on the effectiveness of QAT, we present QAT models starting from RTN and GPTQ in rows 1-4 and rows 5-8, respectively. It can be found that using GPTQ for initialization yields a better starting point, resulting in an average improvement of 42.01% (row 1 vs. row 5), which is also consistent with the findings in Section 3.3.

**The Effect of KD.** Both SFT and KD significantly recover quantization loss. With RTN initialization, SFT yields a 42.64% improvement (row 1 vs. row 2), while KD achieves a 45.09% gain (row 1 vs. row3). Regardless of initialization, the KD approach demonstrates robustly superior performance over SFT. To be specific, KD achieves higher average accuracy than SFT by 2.45% under RTN (row 1 vs. row 2) and by 1.18% under GPTQ (row 3 vs. row 4).

**Further Improvement by GRPO.** To further refine the performance of quantized models trained with knowledge distillation (KD), we integrate GRPO into the training pipeline. It can be seen that GRPO further boosts KD performance by 1.29% under RTN (row 3 vs. row 4) and 2.06% under GPTQ (row 7 vs. row 8), demonstrating its effectiveness in enhancing quantized models through policy refinement. In particular, compared with RTN, GPTQ also provides a better starting point for the RL training.

## A.3 Extensive Experiments on Weight-Activation Quantization

We then examine W4A4KV4 quantization as a representative configuration for weight-activation quantization. This scenario is particularly challenging since weights, activations, and KV cache are quantized to low bits. Note that we implement Reasoning-QAT in this setting by loading the transformation matrices from FlatQuant as initialization and further performing QAT. Unlike the original FlatQuant, which applies layer-wise correction in isolation, our method uses network-wise adjustments during QAT. This holistic optimization makes the model account for the propagation of quantization errors across layers, thereby handling the accumulation of mismatches that single-layer correction cannot capture. As a result, the model can adaptively correct quantization errors in a globally consistent manner rather than relying solely on static PTQ calibration.

**Experiment Settings.** To further validate our method, we evaluate a joint 4-bit weight and 4-bit activation (W4A4) quantization setting. Following the FlatQuant framework, we employ per-channel symmetric quantization for weights and per-token asymmetric quantization for activations. During the KD stage, the model is fine-tuned on 48,000 sequences from the OpenR1-Math dataset, with a fixed sequence length of 8,192. To optimize training stability, we adopt a multi-tier learning rate strategy: standard weights are updated at $1 \times 10^{-6}$, while the transformation matrices and clipping factors in FlatQuant are assigned higher

| | RTN | GPTQ | SFT | KD | GRPO | AIME120 | MATH-500 | GSM8K | AVG |
|---|---|---|---|---|---|---|---|---|---|
| #0 | - | - | - | - | - | 21.67 | 84.40 | 84.61 | 63.56 |
| #1 | ✓ | - | - | - | | 0.83 | 15.00 | 15.39 | 10.41 |
| #2 | ✓ | - | ✓ | - | - | 10.00 | 73.60 | 75.54 | 53.05 |
| #3 | ✓ | - | - | ✓ | - | 14.44 | 76.20 | 75.87 | 55.50 |
| #4 | ✓ | - | - | ✓ | ✓ | 14.44 | 78.00 | 77.93 | 56.79 |
| #5 | - | ✓ | - | - | - | 10.00 | 71.60 | 75.66 | 52.42 |
| #6 | - | ✓ | ✓ | - | - | 14.17 | 75.53 | 76.12 | 55.27 |
| #7 | - | ✓ | - | ✓ | - | 13.89 | 78.20 | 77.26 | 56.45 |
| #8 | - | ✓ | - | ✓ | ✓ | **16.39** | **79.80** | **79.35** | **58.51** |

Table 8: Ablation studies of Reasoning-QAT, including the PTQ initializations (i.e., RTN and GPTQ), QAT training paradigms (i.e., SFT, KD and GRPO) based on R1-Qwen-1.5B.

rates of $5 \times 10^{-5}$ and $5 \times 10^{-4}$, respectively. Configuration for the ensuing Reinforcement Learning (RL) phase remains consistent with the details provided in 7.

**Comparison with Representative PTQ Methods** As shown in 9, PTQ baselines such as QuaRot (Ashkboos et al., 2024) and FlatQuant (Sun et al., 2024) suffer from large performance decreases. Our workflow, however, achieves consistent improvements across all model sizes. For instance, on Qwen3-4B, Reasoning-QAT raises the average score from 58.28% (FlatQuant) to 60.78%, effectively narrowing the gap to full precision and demonstrating that our consolidated workflow can effectively tackle the degradation in W4A4KV4 quantization scenarios and shows the effectiveness of our four key findings. Note that QuaRot results for Qwen3-4B are omitted because its hidden size of 2560 is incompatible with the Hadamard transformation required by the standard kernel.

### A.4 Additional Evaluation on General Domains

**Performance on Weight-Only Setting.** To verify whether our reasoning-focused QAT workflow compromises the model's general capabilities, we evaluated Reasoning-QAT on non-reasoning benchmarks, including HellaSwag (Commonsense) (Zellers et al., 2019), PIQA (Physics) (Bisk et al., 2020), and Winogrande (Commonsense) (Sakaguchi et al., 2021). As shown in Table 10, under the W3G128 setting, Reasoning-QAT maintains performance comparable to the PTQ baseline across all models and tasks, and even achieves slight improvements on PIQA (e.g., +2.45% on R1-Qwen-1.5B). This indicates that while our method is optimized for reasoning, it effectively recovers precision without catastrophic forgetting of general knowledge. We observe similar stability in the W4A4KV4 setting, with detailed results provided in Table 11.

### A.5 Analysis on Initialization for W4A4KV4 QAT

To address concerns regarding the generalizability of our approach and its dependency on specific PTQ methods (e.g., FlatQuant), we conducted additional studies on the W4A4KV4 setting using the R1-Qwen-1.5B model.

**Transferability to Other PTQ Methods (QuaRot).** We investigated whether Reasoning-QAT can transfer to other advanced PTQ methods, such as QuaRot (Ashkboos et al., 2024). As shown in Table 12, while the KD stage successfully transfers and improves performance over the PTQ baseline (e.g., MATH-500 improves from 1.20% to 11.20%), the subsequent RL stage collapses. This empirically validates our insight in Section 3.4 that RL requires a minimum capability threshold to generate rewardable trajectories. Since the QuaRot-based model (for this specific architecture) retained severe outliers even after KD, it failed to provide a viable starting policy for RL. This confirms that while the QAT workflow is transferable, a strong initialization (like FlatQuant) is a prerequisite for the RL stage in W4A4 scenarios.

**Necessity of Outlier Suppression (vs. RTN).** We further justify the use of FlatQuant by comparing it with a standard RTN initialization (i.e., QAT without outlier suppression). As presented in Table 13, applying QAT directly on RTN initialization leads to complete training collapse. This confirms that FlatQuant (or equivalent outlier-suppression

| Model | W-Bits | Methods | AIME-120 | MATH-500 | GSM8K | GPQA-Diamond | LiveCode-Bench | Avg. | Drop ↓ |
|---|---|---|---|---|---|---|---|---|---|
| Qwen3-0.6B | BF16 | - | 11.11 | 74.00 | 79.00 | 28.45 | 12.94 | 41.10 | - |
| | W4A4KV4 | QuaRot | 0.00 | 0.00 | 0.00 | 24.24 | 0.00 | 4.84 | -36.26 |
| | | FlatQuant | 0.28 | 21.67 | 33.06 | 29.80 | 1.87 | 17.34 | -23.76 |
| | | Reasoning-QAT | 0.00 | 30.27 | 48.62 | 26.94 | 1.37 | 21.44 | -19.66 |
| R1-1.5B | BF16 | - | 21.67 | 84.40 | 84.61 | 36.87 | 16.04 | 48.72 | - |
| | W4A4KV4 | QuaRot | 0.00 | 1.20 | 0.76 | 8.59 | 0.00 | 2.11 | -46.61 |
| | | FlatQuant | 10.00 | 64.80 | 78.62 | 31.82 | 6.72 | 38.39 | -10.33 |
| | | Reasoning-QAT | 12.50 | 73.20 | 77.94 | 32.83 | 10.07 | 41.31 | -7.41 |
| Qwen3-4B | BF16 | - | 58.89 | 95.33 | 94.49 | 56.06 | 48.38 | 70.63 | - |
| | W4A4KV4 | FlatQuant | 32.78 | 89.93 | 92.12 | 47.47 | 29.10 | 58.28 | -12.35 |
| | | Reasoning-QAT | 36.67 | 91.40 | 92.42 | 48.48 | 34.95 | 60.78 | -9.85 |

Table 9: Main results of Reasoning-QAT on Qwen3-0.6B, R1-Qwen-1.5B and Qwen3-4B across various reasoning benchmarks.

| Model | Method | HellaSwag | PIQA | Winogrande |
|---|---|---|---|---|
| Qwen3-0.6B | GPTQ (Baseline) | 29.35 | 59.36 | **50.28** |
| | **Reasoning-QAT** | **29.41** | **59.79** | 50.19 |
| R1-Qwen-1.5B | GPTQ (Baseline) | 32.85 | 60.01 | **51.22** |
| | **Reasoning-QAT** | **33.96** | **62.46** | 50.71 |
| Qwen3-4B | GPTQ (Baseline) | 42.83 | 69.10 | **53.07** |
| | **Reasoning-QAT** | **42.95** | **70.13** | 53.00 |

Table 10: Comparison of Reasoning-QAT and GPTQ on general domain benchmarks under W3G128 quantization. Our method maintains robust performance on non-reasoning tasks.

techniques) is not a confounding variable but a *necessary precondition* to condition the optimization landscape for stable W4A4 training.

### A.6 Computational Budget and Infrastructure

All experiments were conducted on a compute cluster equipped with high-performance NVIDIA GPUs. The training for the 1.5B model required approximately 96 GPU hours, and the 4B model required approximately 145 GPU hours.

### A.7 Artifacts and Licenses

We utilize the following scientific artifacts, strictly adhering to their respective licenses and terms of use.

**Models**

- DeepSeek-R1-Distill-Qwen-1.5B: Released under the MIT License.
- Qwen3 Family (0.6B and 4B): Released under the Apache 2.0 License.

**Datasets**

- OpenR1-Math: Used for training and distillation, released under the Apache 2.0 License.
- Wikitext-2: Used for calibration comparison, released under the Creative Commons Attribution-ShareAlike (CC BY-SA) License.
- Evaluation Benchmarks: We use standard evaluation datasets including GSM8K, MATH-500, AIME-120, GPQA-Diamond, and LiveCodeBench. These are publicly available and used in accordance with their original licenses (typically MIT or Apache 2.0) for research evaluation purposes.

We confirm that all artifacts were used consistent with their intended use. We verified that the data subsets used do not contain personally identifying information (PII) or offensive content.

### A.8 Potential Risks and Ethical Considerations

Our work focuses on the quantization of reasoning models to improve inference efficiency. While the quantized models generally retain the capabilities of their full-precision counterparts, they may inherit biases or hallucinations inherent in the base large language models (LLMs). Our methods do not introduce additional safety risks beyond those already present in the pre-trained backbones.

| Model | Method | HellaSwag | PIQA | Winogrande |
|---|---|---|---|---|
| Qwen3-0.6B | FlatQuant (Baseline) | 34.39 | 62.24 | 52.05 |
| | **Reasoning-QAT** | **34.42** | **62.40** | **52.13** |
| R1-Qwen-1.5B | FlatQuant (Baseline) | 34.75 | 61.26 | **51.38** |
| | **Reasoning-QAT** | **34.78** | **62.35** | 50.32 |
| Qwen3-4B | FlatQuant (Baseline) | 44.94 | 69.10 | 54.10 |
| | **Reasoning-QAT** | **46.03** | **70.35** | **54.30** |

Table 11: Comparison of Reasoning-QAT and FlatQuant on general domain benchmarks under W4A4KV4 quantization.

| Initialization | Method | AIME-120 | MATH-500 | GSM8K | GPQA-Diamond | LCB | Status |
|---|---|---|---|---|---|---|---|
| QuaRot | PTQ | 0.00 | 1.20 | 0.76 | 8.59 | 0.00 | Collapsed |
| | + KD | 1.67 | 11.20 | 12.74 | 9.09 | 1.12 | Improved |
| | + KD + RL | 0.00 | 3.60 | 6.07 | 2.53 | 0.00 | Collapsed |
| FlatQuant | PTQ | 10.00 | 64.80 | 78.62 | 31.82 | 6.72 | Converged |
| | + KD + RL | **12.50** | **73.20** | **77.94** | **32.83** | **10.07** | **Converged** |

Table 12: Investigating the transferability of Reasoning-QAT using QuaRot initialization on R1-Qwen-1.5B (W4A4KV4). While KD improves performance, the model fails to sustain RL training due to insufficient base capability.

### A.9 AI Writing Assistance

We utilized AI assistants strictly for grammatical error correction and text polishing. All scientific claims, experimental designs, and results presented in this paper were verified by the authors.

| Initialization | Method | AIME-120 | MATH-500 | GSM8K | GPQA-Diamond | LCB | Status |
|---|---|---|---|---|---|---|---|
| RTN | KD | 0.00 | 0.60 | 1.29 | 4.55 | 0.00 | Collapsed |
| | Reasoning-QAT | 0.00 | 0.20 | 0.37 | 1.01 | 0.00 | Collapsed |
| FlatQuant | Reasoning-QAT | **12.50** | **73.20** | **77.94** | **32.83** | **10.07** | **Converged** |

Table 13: Comparison of RTN vs. FlatQuant initialization for W4A4KV4 QAT on R1-Qwen-1.5B. Without outlier suppression (FlatQuant), QAT faces complete collapse.