BEYOND FULL IMITATION: DIRECTION-GUIDED QUANTIZATION-AWARE ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Microscaling floating-point (MXFP) formats have emerged as a promising low-precision representation for efficient large language model (LLM) training and inference, yet quantizing both weights and activations to 4 bits (also known as W4A4 quantization) can still severely degrade performance on complex reasoning tasks. While existing post-training quantization (PTQ) and quantization-aware distillation (QAD) methods struggle to fully recover BF16 performance under aggressive quantization, on-policy distillation (OPD) is naturally suited to this setting because it supervises the quantized student on its own generated trajectories and better matches the states encountered during inference. However, we find that full imitation of the BF16 teacher is unnecessary for effective recovery. Quantization can either weaken or strengthen preferences introduced by post-training, suggesting that not every deviation from the BF16 teacher should be corrected. Based on this insight, we introduce DQ-OPD, a **D**irection-guided **Q**uantization-aware **O**n-**P**olicy **D**istillation method that selectively corrects preference-weakening deviations while preserving preference-strengthening ones. Across Qwen3-8B and Qwen3-30B-A3B under MXFP W4A4 quantization, DQ-OPD consistently outperforms all PTQ and QAD baselines. On Qwen3-30B-A3B, it narrows the average gap to BF16 to 0.71 points and even surpasses the BF16 teacher on AIME24. These results demonstrate that near-BF16 performance under W4A4 quantization can be achieved without fully imitating the BF16 teacher, highlighting the importance of deciding which quantization deviations to be corrected rather than applying teacher correction uniformly.

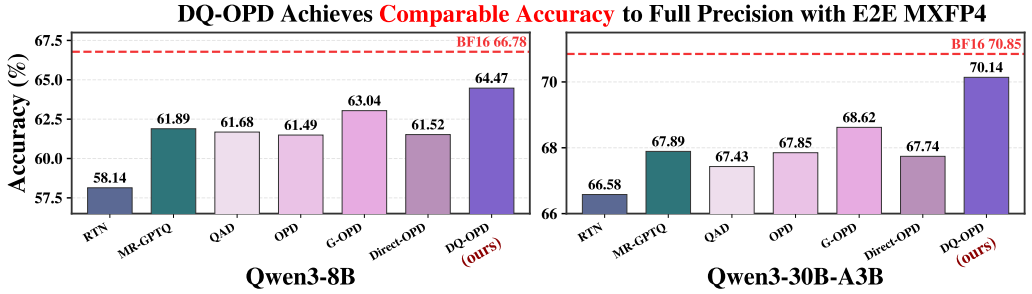


Figure 1: Overall performance across seven benchmarks of different methods on Qwen3-8B (*left*) and Qwen3-30B-A3B (*right*) under MXFP4 W4A4 quantization. DQ-OPD achieves the strongest W4A4 results on both scales, narrowing the gap to BF16 on Qwen3-30B-A3B to 0.71 points.

1 INTRODUCTION

Large language models (LLMs) have rapidly increased in both scale and computational demand (Liu et al., 2024a; Glm et al., 2024; Yang et al., 2025; Guo et al., 2025; Team et al., 2026; Xiao et al., 2026), making efficient numerical representations increasingly important for reducing the memory, computation, and energy costs of training and inference. With growing hardware support for

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.

low-precision arithmetic, microscaling floating-point (MXFP) formats have emerged as a promising solution for modern LLM workloads (Rouhani et al., 2023; Open Compute Project, 2023). By combining fine-grained shared scaling with floating-point representations, MXFP preserves numerical dynamic range at ultra-low precision while remaining hardware efficient. In particular, MXFP has been increasingly adopted in large-scale training and inference systems (Tseng et al., 2025; Cim et al., 2026; Agarwal et al., 2025; Xu et al., 2026). Nevertheless, aggressively quantizing both weights and activations to 4 bits (i.e., W4A4 quantization) still incurs huge accuracy degradation, especially on reasoning tasks and long chain-of-thought generation.

A large body of post-training quantization (PTQ) methods has been developed to mitigate this degradation without additional training (Xiao et al., 2023; Ashkboos et al., 2024; Egiazarian et al., 2026; Li et al., 2026a; Chhugani et al., 2026; Liu et al., 2026a; Zhang et al., 2026b). However, existing PTQ methods still struggle to recover BF16 performance under aggressive MXFP W4A4 quantization (Zhang et al., 2026a). Quantization-aware distillation (QAD) offers a strong alternative by optimizing the quantized student under quantization noise while aligning it with a full-precision teacher. Yet QAD still fails to close the gap to BF16 under low-bit weight-activation quantization (Egiazarian et al., 2026). A key limitation is that QAD is typically trained on fixed teacher-generated or offline trajectories, while quantization alters the student's token distribution during autoregressive generation. As a result, the quantized student may visit contexts poorly covered by the distillation data, creating a training-inference mismatch that accumulates over long chains of thought.

Recent progress in on-policy distillation (OPD) provides a natural way to address this state-distribution mismatch (Gu et al., 2024; Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025; Yang et al., 2025; Li et al., 2026b). Unlike conventional distillation that relies on fixed teacher-generated trajectories, OPD performs teacher supervision directly on trajectories sampled from the current student policy, enabling optimization on the states actually visited during inference. This property makes OPD a natural framework for addressing compounding quantization-induced deviations during long-horizon generation. Furthermore, from the perspective of reinforcement learning, OPD can be interpreted as providing dense token-level supervision, which enables more flexible objective design beyond strict teacher imitation (Yang et al., 2026; Feng et al., 2026).

In this work, we investigate quantization-aware on-policy distillation for aggressively quantized LLMs and uncover a counterintuitive phenomenon: fully imitating the BF16 teacher is not necessary to recover the performance lost to quantization. Our analysis shows that quantization can either weaken or strengthen the preferences introduced by post-training, implying that not every deviation from the BF16 teacher should be treated as an error. Motivated by this finding, we introduce DQ-OPD, a quantization-aware on-policy distillation framework that selectively applies teacher correction according to the relationship between the post-training and quantization deviations. By correcting deviations that weaken post-training preferences while preserving those that strengthen them, DQ-OPD recovers performance loss induced by quantization without fully imitating teacher.

We evaluate DQ-OPD on Qwen3-8B and Qwen3-30B-A3B across seven benchmarks spanning knowledge, coding, mathematical, and scientific reasoning under MXFP4 W4A4 quantization. While RTN suffers from significant degradation and standard OPD still lags behind BF16 performance, DQ-OPD achieves the strongest overall W4A4 results on both model scales. Notably, on Qwen3-30B-A3B, it narrows the average gap to BF16 to merely 0.71 points and even surpasses the BF16 baseline on AIME24. Our main contributions are summarized as follows:

- **Rethinking teacher imitation for quantized models.** We show that deviations introduced by quantization are not uniformly harmful. Quantization can either weaken or strengthen preferences learned during post-training. This reveals that recovering a quantized model does not necessarily require restoring its entire BF16 behavior.
- **Direction-guided quantization-aware on-policy distillation.** We propose DQ-OPD, a direction-guided on-policy distillation framework that supervises quantized student trajectories. By leveraging directional signals to distinguish deviation types, DQ-OPD selectively corrects preference-weakening deviations while preserving preference-strengthening ones.
- **State-of-the-art MXFP4 W4A4 performance.** Across Qwen3-8B and Qwen3-30B-A3B, DQ-OPD achieves the best overall performance among the evaluated PTQ and QAD methods. On the 30B model, it further reduces the average gap between W4A4 and BF16 precisions to only 0.71 points and slightly exceeds BF16 performance on AIME24.

2 Background and Related Work

Microscaling Floating-Point Format. Microscaling floating-point (MXFP) formats have recently emerged as a hardware-efficient alternative to conventional integer quantization for low-precision LLM training and inference (Rouhani et al., 2023; Open Compute Project, 2023; Chen et al., 2025b). MXFP employs fine-grained block-wise scaling, where a small group of elements shares a scale factor, enabling compact floating-point representations while retaining a wider dynamic range. In particular, MXFP4 represents each element in FP4 (E2M1) and shares an E8M0 scale across every 32 elements. The use of power-of-two E8M0 scales simplifies hardware implementation.

Post-Training Quantization. Post-training quantization (PTQ) compresses pretrained models to low precision using limited calibration data without end-to-end retraining. Existing methods control quantization error in different ways. GPTQ uses approximate second-order information for weight quantization (Frantar et al., 2022), while AWQ and SmoothQuant redistribute numerical ranges through channel scaling (Lin et al., 2024b; Xiao et al., 2023). Transformation-based methods introduce broader degrees of freedom through learned transformations (Shao et al., 2024; Ashkboos et al., 2024; Liu et al., 2025; Lin et al., 2024a; Sun et al., 2024). Recent methods adapt these principles to microscaling formats through format-aware reconstruction, block-wise transformations, and fine-grained scaling (Egiazarian et al., 2026; Li et al., 2026a; Chhugani et al., 2026). These advances narrow the quantization gap. Nevertheless, quantizing both weights and activations to 4 bits remains challenging. A recent benchmark finds MXFP8 nearly lossless but reports substantial accuracy loss under MXFP4 (Zhang et al., 2026a). This gap motivates optimization beyond calibration-only PTQ.

Quantization-Aware Training and Distillation. Quantization-aware training (QAT) directly optimizes model parameters under simulated low-precision computation, providing greater flexibility than post-training quantization in challenging low-bit regimes and complex downstream tasks (Chen et al., 2025a; Jeon et al., 2025; Liu et al., 2026b; Lee et al., 2026). Quantized training can also be coupled with knowledge distillation from a full-precision teacher. This line of work has progressed from compact neural networks (Polino et al., 2018; Mishra & Marr, 2017) and quantized Transformers (Bai et al., 2021; Kim et al., 2022) to generative LLMs, including LLM-QAT and BitDistiller (Liu et al., 2024b; Du et al., 2024). More recent studies extend quantization-aware distillation (QAD) to reasoning models and hardware-native FP4 formats (Lv et al., 2026; Xin et al., 2026; Egiazarian et al., 2026). Despite these advances, W4A4 quantization with MXFP format which is generally more challenging than NVFP4 (Chhugani et al., 2026), still leaves a noticeable gap to BF16.

On-Policy Distillation. Conventional knowledge distillation transfers teacher predictions on a fixed data distribution (Hinton et al., 2015), whereas on-policy distillation (OPD) trains on student-generated trajectories and queries the teacher on the states actually visited by the student (Agarwal et al., 2024; Gu et al., 2024; Lu & Thinking Machines Lab, 2025; Heo et al., 2026; Yang et al., 2025). By reducing the distribution mismatch between distillation and inference, OPD has shown strong effectiveness in post-training and long-horizon generation. G-OPD further interprets OPD as KL-regularized reinforcement learning and introduces flexible reference policies and reward scaling (Yang et al., 2026), while Direct-OPD constructs the reward from paired pre- and post-RL teacher checkpoints to transfer post-training improvements (Feng et al., 2026). Compared with offline QAD, OPD is appealing for quantized models because supervision is applied on trajectories generated by the quantized student itself, allowing quantization-induced deviations to be corrected as they emerge during autoregressive generation. Our work builds on this on-policy formulation and studies how such teacher corrections should be selectively applied for performance recovery.

3 Do Quantized Models Need to Fully Imitate the BF16 Teacher?

OPD provides a natural framework for recovering the performance of aggressively quantized reasoning models. Let π_T denote a post-trained BF16 teacher, π_Q its W4A4 quantized counterpart, and π_θ a trainable W4A4 policy initialized from π_Q. Standard OPD minimizes the reverse KL divergence between the student and the teacher on student-generated trajectories (Gu et al., 2024; Lu & Thinking Machines Lab, 2025). For prompts $x \sim \mathcal{D}$ and responses $y = (y_1, \ldots, y_L)$ sampled from the current student policy, the objective can be written as

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \| \pi_T(\cdot \mid x)) \,. \tag{1}$$

Using the quantized initialization π_Q as the reference policy, the same objective admits the following KL-regularized reinforcement-learning interpretation:

$$\mathcal{J}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x)}\left[\log\frac{\pi_T(y \mid x)}{\pi_Q(y \mid x)}\right] - \mathbb{E}_{x\sim\mathcal{D}} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x)\|\pi_Q(\cdot \mid x))\,. \tag{2}$$

The first term provides an implicit dense reward that encourages the quantized policy to move toward the BF16 teacher, while the second regularizes the policy toward its W4A4 initialization. In practice, however, standard OPD still does not fully close the gap between W4A4 students and their BF16 teachers, as shown in Tables 1 and 2. This observation raises a more fundamental question: *is full teacher imitation actually necessary for recovering the performance lost to quantization?* To investigate this question, we introduce the BF16 base policy π_B, from which the post-trained teacher π_T is obtained. The transition from π_B to π_T captures the policy change introduced by post-training, whereas the transition from π_Q to π_T reflects the teacher correction induced by quantization. Comparing these two directions allows us to determine whether quantization weakens or strengthens the change learned during post-training.

For a prefix $s_t = (x, y_{<t})$ and candidate token a, we measure the post-training shift as

$$\Delta_{\mathrm{PT}}(a \mid s_t) = \log\frac{\pi_T(a \mid s_t)}{\pi_B(a \mid s_t)}, \tag{3}$$

which captures the direction in which post-training changes the preference for token a. We similarly measure the teacher correction induced by quantization as

$$\Delta_{\mathrm{Q}}(a \mid s_t) = \log\frac{\pi_T(a \mid s_t)}{\pi_Q(a \mid s_t)}. \tag{4}$$

Their sign relationship distinguishes two qualitatively different effects of quantization:

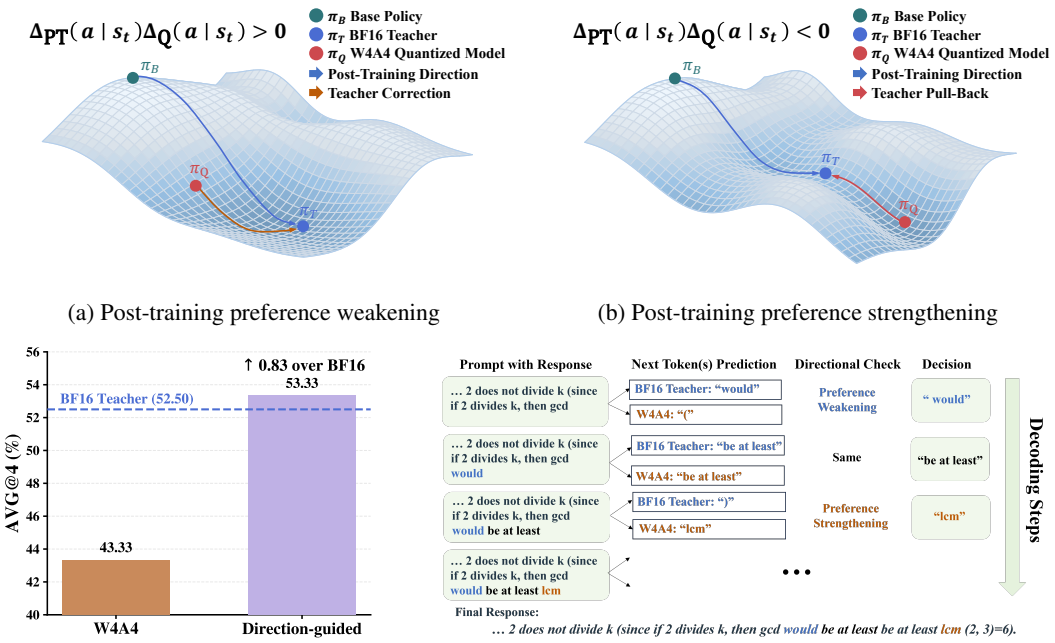
- **Preference weakening.** As shown in Figure 2a, when $\Delta_{\mathrm{PT}}(a \mid s_t)\Delta_{\mathrm{Q}}(a \mid s_t) > 0$, quantization moves the token probability against the direction induced by post-training, thereby compromising the learned preference. In this case, correcting the quantized policy toward the BF16 teacher restores the post-training effect compromised by quantization.
- **Preference strengthening.** As shown in Figure 2b, when $\Delta_{\mathrm{PT}}(a \mid s_t)\Delta_{\mathrm{Q}}(a \mid s_t) < 0$, quantization moves the token probability further along the post-training direction, thereby strengthening the learned preference. Although such deviations increase the discrepancy from the BF16 teacher, they do not necessarily represent harmful quantization error. Pulling them back toward the teacher may instead undo a policy shift that is consistent with post-training direction.

These two cases motivate our central hypothesis: *quantization recovery should correct preference weakening while preserving preference strengthening*. We test this hypothesis through a direction-guided decoding intervention on Qwen3-8B.

Direction-guided decoding intervention. We visualize the decoding steps on Qwen3-8B in Figure 2d. When the W4A4 policy π_Q and the BF16 teacher π_T propose different tokens, we examine whether the quantization-induced change for the W4A4 candidate weakens the preference introduced by post-training. If so, we replace the quantized choice with the teacher token. Otherwise, we retain the quantized token. This intervention therefore corrects preference weakening while preserving preference strengthening, without performing any parameter update. On HMMT Nov. 2025 (Dekoninck et al., 2026) (30 problems, 4 samples per problem), the original W4A4 model achieves 43.33, compared with 52.50 for the BF16 teacher. Direction-guided decoding improves the W4A4 model to 53.33, closing the entire quantization gap and surpassing the BF16 teacher by 0.83 points (Figure 2c). This controlled experiment provides direct evidence that *recovering BF16 performance does not require full imitation of BF16 behavior*. Instead, selectively correcting preference weakening while preserving preference strengthening is sufficient to restore, and in this setting slightly exceed, teacher-level performance.

4 Direction-Guided Quantization-Aware On-Policy Distillation

The analysis in Section 3 shows that quantization can either weaken or strengthen preferences induced by post-training. This distinction suggests that teacher correction should not be applied uniformly. Preference-weakening deviations should be corrected toward the BF16 teacher, whereas



(a) Post-training preference weakening

(b) Post-training preference strengthening

(c) Direction-guided correction fully recovers performance lost to quantization

(d) Token-level intervention updates the rollout context sequentially

Figure 2: **Direction-guided decoding intervention on Qwen3-8B.** The geometry of preference weakening and preference strengthening is illustrated in (a) and (b), respectively. Accuracy on HMMT Nov. 2025 is reported in (c), and (d) illustrates how the resulting token-level decisions shape subsequent decoding.

preference-strengthening deviations can be preserved. Motivated by this observation, we introduce a directional mask to the OPD correction target, retaining only the necessary corrections.

4.1 Direction-Guided Objective

At a prefix $s_t = (x, y_{<t})$, we use the post-training preference shift $\Delta_{\mathrm{PT}}(a \mid s_t)$ and the quantization deviation $\Delta_{\mathrm{Q}}(a \mid s_t)$ defined in Eq. (3) and Eq. (4) to determine whether a teacher correction should be retained for a candidate token a. Specifically, a preference weakening deviation satisfies

$$M(a \mid s_t) = \mathbf{1}[\Delta_{\mathrm{PT}}(a \mid s_t)\Delta_{\mathrm{Q}}(a \mid s_t) > 0]\,. \tag{5}$$

For teacher corrections satisfying this criterion, we preserve the original OPD implicit reward:

$$r_A(a \mid s_t) = M(a \mid s_t) \log \frac{\pi_T(a \mid s_t)}{\pi_Q(a \mid s_t)} = M(a \mid s_t)\Delta_{\mathrm{Q}}(a \mid s_t). \tag{6}$$

Therefore, the directional criterion determines *whether* a teacher correction is applied, while the correction itself retains the magnitude and direction of the standard OPD reward. Preference-weakening deviations receive the original teacher correction, whereas preference-strengthening deviations are left uncorrected. For a response $y = (y_1, \ldots, y_L)$, we accumulate the selected token rewards along the prefixes visited by that response:

$$R_A(y \mid x) = \sum_{t=1}^{L} r_A(y_t \mid s_t), \qquad s_t = (x, y_{<t}). \tag{7}$$

This sequence reward yields the directionally selective OPD objective

$$\mathcal{J}_\lambda(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}[\lambda R_A(y \mid x)] - \mathbb{E}_{x\sim\mathcal{D}} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \| \pi_Q(\cdot \mid x))\,. \tag{8}$$

The first term rewards the teacher corrections selected by Eq. (6). The second retains the W4A4 initialization as the reference policy. Motivated by the reward-extrapolation view of G-OPD (Yang et al., 2026), we use $\lambda > 0$ to control the strength of the selected reward.

4.2 IDEALIZED TARGET POLICY

We next characterize the target policy induced by the direction-guided reward. For a fixed prompt x, let π_A denote the unrestricted optimum of Eq. (8) at unit reward scale ($\lambda = 1$), whose closed-form solution is derived in Appendix A.2. As derived in Appendix A.3, its autoregressive factorization is

$$\pi_A(y \mid x) \propto \prod_{t=1}^{L} \pi_T(y_t \mid s_t)^{M_t} \pi_Q(y_t \mid s_t)^{1-M_t}, \qquad M_t = M(y_t \mid s_t). \tag{9}$$

This factorization makes the effect of selective correction explicit: for preference-weakening deviations, the target adopts the corresponding BF16 teacher factor, whereas preference-strengthening deviations retain the W4A4 factor. Eq. (9) therefore characterizes how token-level reward selection induces a response-level target distribution. For a general reward scale $\lambda > 0$, the idealized optimum of Eq. (8) derived in Appendix A.4 becomes

$$\pi^*_\lambda(y \mid x) \propto \pi_Q(y \mid x) \exp(\lambda R_A(y \mid x)) \propto \pi_A(y \mid x)^\lambda \pi_Q(y \mid x)^{1-\lambda}. \tag{10}$$

This relation separates reward selection from reward strength. The mask M determines which teacher corrections shape the target, while λ controls its displacement from π_Q along the corresponding response log-odds path. Accordingly, $0 < \lambda < 1$ interpolates between π_Q and π_A, $\lambda = 1$ recovers π_A, and $\lambda > 1$ extrapolates beyond π_A along the same log-odds direction.

4.3 ON-POLICY OPTIMIZATION

Eq. (8) assigns credit across the tokens of a sampled response. Following the zero-discount approximation used in OPD (Yang et al., 2026), we instead update each sampled token using its immediate contribution. Appendix A.5 derives the exact gradient and this approximation. For $y \sim \pi_\theta(\cdot \mid x)$, the coefficient at step t is

$$\widehat{A}_t = \mathrm{sg}\left[\lambda r_A(y_t \mid s_t) - \log \frac{\pi_\theta(y_t \mid s_t)}{\pi_Q(y_t \mid s_t)}\right], \tag{11}$$

where sg denotes stop gradient. The first term supplies the directionally selected teacher reward, and the second is the local contribution of the reference-policy regularizer. We use $\widehat{A}_t$ as the coefficient of the sampled-token policy-gradient update,

$$\nabla_\theta \mathcal{J}_\lambda(\theta) \approx \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{L} \widehat{A}_t \nabla_\theta \log \pi_\theta(y_t \mid s_t)\right]. \tag{12}$$

The target-policy analysis above characterizes the ideal sequence-level objective, while Eq. (12) gives the local on-policy surrogate used for optimization. The training procedure is in Algorithm 1.

5 EXPERIMENTS

We evaluate DQ-OPD across four aspects: (i) overall performance recovery, (ii) direction-guided correction, (iii) KL regularization for preference-strengthening deviations, and (iv) reward scaling. Below, we detail the experimental settings and results.

5.1 EXPERIMENTAL SETUP

Quantization Configuration. Our experiments are conducted on a dense Qwen3-8B model and a sparse Qwen3-30B-A3B Mixture-of-Experts (MoE) model (Yang et al., 2025). For each model, we denote the BF16 base policy as π_B, its post-trained BF16 checkpoint as the teacher π_T, and the quantized version of π_T as π_Q. We apply MXFP4 W4A4 quantization to the weights and activations of all linear layers in the Transformer blocks, including the QKV and output projections in self-attention and the gate, up, and down projections in dense MLPs and MoE experts. The MoE router remains in BF16. All main experiments use MBS with 128-element sharing at an E0M8 macro scale (Chhugani et al., 2026). We also provide results without MBS in Appendix C.1.

Training Configuration. We train Qwen3-8B and Qwen3-30B-A3B with VERL (Sheng et al., 2025), using FSDP2 for optimization and vLLM for rollout generation. Training prompts come

Table 1: **Main results on Qwen3-8B under MXFP4 W4A4 quantization.** Scores are percentages. *Avg.* is the unweighted mean across seven benchmarks, and *Recovery* normalizes this mean by the BF16 result. The best quantized result in each column is shown in **bold**.

Type	Method	Knowledge	Code			Math		Science	Summary	
		MMLU-Pro	LCB v6	MBPP+	HE+	AIME24	AIME25	GPQA	Avg.	Recovery
	BF16	73.77	43.27	74.21	83.53	71.46	63.12	58.13	66.78	100.00
PTQ	RTN	69.94	32.99	68.45	79.12	57.08	47.29	52.12	58.14	87.06
	MR-GPTQ	70.20	**38.58**	71.92	80.59	64.17	53.54	54.20	61.89	92.67
QAT	QAD	**70.37**	36.42	70.97	81.09	64.38	54.17	54.34	61.68	92.36
	OPD	68.74	34.86	71.03	79.96	64.58	57.08	54.19	61.49	92.08
	G-OPD	67.95	37.46	71.56	**83.80**	67.08	58.75	54.65	63.04	94.40
	Direct-OPD	69.53	34.93	69.31	80.79	67.29	54.79	53.99	61.52	92.12
	DQ-OPD	69.82	37.77	**73.08**	82.77	**70.83**	**60.63**	**56.36**	**64.47**	**96.54**

Table 2: **Main results on Qwen3-30B-A3B under MXFP4 W4A4 quantization.** Scores are percentages. *Avg.* is the unweighted mean across seven benchmarks, and *Recovery* normalizes this mean by the BF16 result. The best quantized result in each column is shown in **bold**.

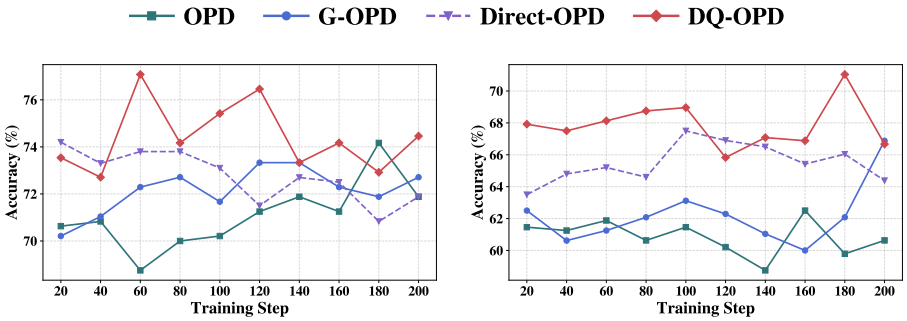
Type	Method	Knowledge	Code			Math		Science	Summary	
		MMLU-Pro	LCB v6	MBPP+	HE+	AIME24	AIME25	GPQA	Avg.	Recovery
	BF16	77.48	48.10	76.65	85.98	75.00	69.79	62.98	70.85	100.00
PTQ	RTN	74.82	42.44	75.73	83.69	71.88	58.13	59.39	66.58	93.97
	MR-GPTQ	74.73	46.18	76.06	84.76	72.50	60.83	60.20	67.89	95.82
QAT	QAD	74.88	43.46	75.53	85.06	71.04	61.04	60.96	67.43	95.16
	OPD	74.13	44.05	76.26	85.21	74.17	59.79	61.31	67.85	95.77
	G-OPD	74.14	**46.28**	75.53	84.76	72.71	66.88	60.05	68.62	96.85
	Direct-OPD	71.90	43.70	73.68	**86.74**	73.80	65.20	59.20	67.74	95.61
	DQ-OPD	**75.87**	**46.28**	**76.32**	85.67	**75.42**	**68.96**	**62.47**	**70.14**	**99.00**

from OpenThoughts3-1.2M (Guha et al., 2026). QAD uses the responses provided by the dataset, whereas OPD and its variants generate 1 response per prompt with temperature 1.0 and top-p 1.0. Each update contains 256 prompts. Prompts and responses are limited to 2,048 and 8,192 tokens, respectively. We train for 200 steps with a learning rate of 1e-5 without warmup. We set the reward scale to $\lambda = 1.25$ for Qwen3-8B and $\lambda = 4$ for Qwen3-30B-A3B. More details are in Appendix D.

Evaluation Configuration. We evaluate mathematical reasoning on AIME 2024 (Zhang & Math-AI, 2024) and AIME 2025 (Zhang & Math-AI, 2025), scientific reasoning on GPQA Diamond (Rein et al., 2023), general knowledge on MMLU-Pro (Wang et al., 2024), and code generation on LiveCodeBench v6 (Jain et al., 2025), MBPP+, and HumanEval+ (Liu et al., 2023). AIME scores are averaged over 16 samples per problem. GPQA reports pass@1 averaged over 10 samples. Besides, the three code benchmarks report pass@1 averaged over 4 samples. We set the temperature to 0.6 and top-p to 0.95 for evaluation. We compare DQ-OPD with round-to-nearest quantization (RTN), MR-GPTQ (Egiazarian et al., 2026), quantization-aware distillation (QAD), standard OPD (Agarwal et al., 2024; Gu et al., 2024), G-OPD (Yang et al., 2026), and Direct-OPD (Feng et al., 2026).

5.2 Main Results

Tables 1 and 2 present the main results under MXFP4 W4A4 quantization. At both model scales, DQ-OPD achieves the best overall performance among all evaluated quantization methods, recovering 96.54% and 99.00% of BF16 performance on Qwen3-8B and Qwen3-30B-A3B, respectively, and consistently outperforming PTQ and distillation-based baselines. On Qwen3-8B, DQ-OPD achieves an average score of 64.47, outperforming the strongest baseline, G-OPD, by 1.43 points.



(a) AIME 2024 (b) AIME 2025

Figure 3: **AIME accuracy across training steps on Qwen3-30B-A3B.** DQ-OPD outperforms baseline methods at most training steps, sustaining its advantage throughout training.

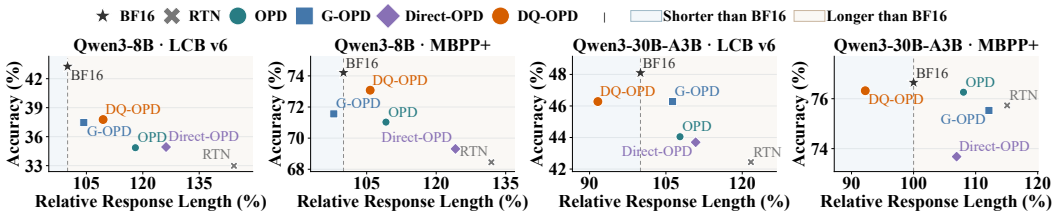


Figure 4: **Downstream accuracy vs. average response length.** Response length is normalized by the BF16 model (dashed line), with shaded regions indicating shorter or longer responses. Distillation removes most length distortion from W4A4 quantization, yet accuracy does not depend on matching the BF16 length.

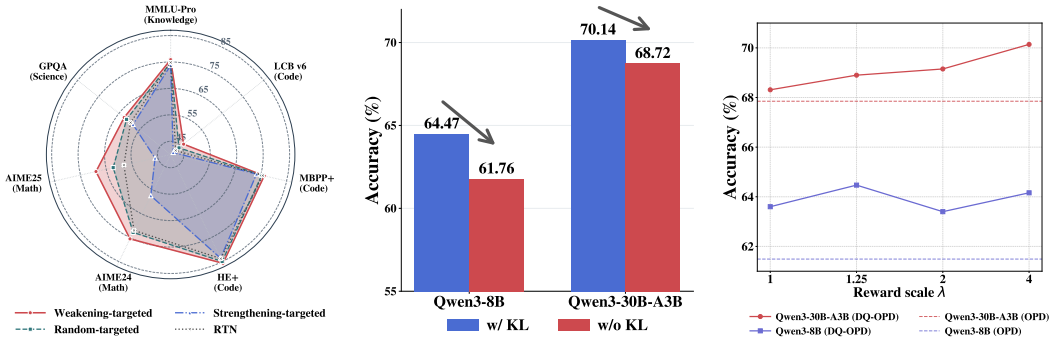
This advantage also generalizes from dense to Mixture-of-Experts (MoE) architectures: on Qwen3-30B-A3B, DQ-OPD reaches 70.14, 1.52 points above G-OPD, leaving a gap of only 0.71 points to the BF16 model. The gains are especially large on complex reasoning: DQ-OPD improves AIME 2024 and AIME 2025 to 70.83 and 60.63 on Qwen3-8B, and to 75.42 and 68.96 on Qwen3-30B-A3B. Notably, on AIME 2024, the 30B model slightly outperforms the BF16 teacher itself (75.42 vs. 75.00), suggesting that effective performance recovery can go beyond exact teacher imitation.

Figure 4 provides additional evidence. While OPD methods restore the response length distorted severely by RTN, a clear accuracy gap remains, which underscores the necessity of direction-guided correction. The training curves in Figure 3 demonstrate a consistent trend. Specifically, DQ-OPD maintains higher AIME scores than the baselines at most training steps. Beyond mathematical reasoning, the gains also extend to knowledge, coding, and scientific tasks: DQ-OPD achieves the best quantized results on MBPP+ and GPQA for Qwen3-8B, and on LCB v6, MMLU-Pro, and MBPP+ for Qwen3-30B-A3B, while remaining competitive elsewhere. This shows that direction-guided teacher correction improves general model capabilities rather than overfitting to specific benchmarks. Overall, DQ-OPD delivers the strongest W4A4 performance among the evaluated PTQ and QAD methods, with near-lossless recovery on Qwen3-30B-A3B, demonstrating the effectiveness of direction-guided correction under aggressive low-bit quantization.

5.3 Ablation Studies

We study three questions underlying our objective: (i) whether preference weakening provides an effective criterion for applying teacher corrections, (ii) whether the KL constraint is important when teacher correction is withheld, and (iii) how the strength of the OPD reward affects performance.

Which deviations should receive teacher correction? Our method applies OPD correction when quantization weakens a post-training preference, identified by $\Delta_{\mathrm{PT}}(a \mid s_t)\Delta_{\mathrm{Q}}(a \mid s_t) > 0$. Figure 5a compares this criterion against two controls. **Random correction** corrects the same fraction of tokens but selects them at random. **Strengthening-targeted correction** instead corrects deviations that strengthen post-training preferences: $\Delta_{\mathrm{PT}}(a \mid s_t)\Delta_{\mathrm{Q}}(a \mid s_t) < 0$. Correcting preference-



(a) Effect of direction-guided correction on Qwen3-30B-A3B.

(b) Effect of KL regularization on tokens whose quantization deviations strengthen preference.

(c) Ablation of reward scale λ on different models.

Figure 5: **Ablation studies of DQ-OPD.** (a) Correcting preference-weakening deviations substantially outperforms both random correction and correcting preference-strengthening deviations, validating the direction-guided design. (b) Retaining KL regularization toward the quantized reference policy π_Q for preference-strengthening deviations improves performance. (c) Our method outperforms standard OPD across all tested values of λ, indicating robustness to the correction strength.

weakening deviations consistently outperforms both controls, while random correction performs worse despite the same correction rate. More importantly, correcting preference-strengthening deviations causes substantial degradation. These results show that teacher correction should target preference-weakening deviations, while preference-strengthening ones should be preserved.

Is KL regularization necessary for preference-strengthening deviations? Our objective does not apply teacher correction when quantization strengthens the post-training preference, but still retains the KL regularization toward the quantized reference policy π_Q. To examine whether this regularization is necessary, we compare against a variant that removes the KL regularization for the tokens whose quantization deviations strengthen preference. As shown in Figure 5b, retaining the KL term consistently improves performance, indicating that staying close to the quantized reference remains important for preference-strengthening deviations. Detailed results are in Appendix C.3.

How strong should the teacher correction be? The reward scale λ controls the strength of the OPD correction while leaving the correction criterion and the KL term toward π_Q unchanged. As shown in Figure 5c, direction-guided correction consistently outperforms standard OPD across all tested values of λ, with relatively stable performance over the tested range. Notably, the method already surpasses the OPD baseline at the mildest setting, $\lambda = 1$, suggesting that the gains stem from distinguishing preference-strengthening from preference-weakening deviations. Detailed benchmark results are reported in Appendix C.2.

6 Conclusion

In this work, we address performance recovery for LLMs under aggressive MXFP4 W4A4 quantization through on-policy quantization-aware distillation. Rather than treating every deviation from the BF16 teacher as an error, we show that quantization can either weaken or strengthen post-training preferences, and that teacher correction should therefore be applied selectively. Building on this insight, we propose DQ-OPD, a direction-guided, on-policy quantization-aware distillation framework that corrects only the deviations that matter. On Qwen3-8B and Qwen3-30B-A3B, DQ-OPD achieves the strongest overall MXFP4 W4A4 performance among evaluated PTQ and QAD methods, narrowing the average gap to BF16 to 0.71 points on Qwen3-30B-A3B and even surpassing the BF16 teacher on AIME24. More broadly, our results suggest that for quantized models, deciding which deviations to correct is a key part of effective recovery. Future work could examine whether direction-guided correction generalizes to other low-precision formats, such as NVFP4, and to longer-horizon agentic settings with environment feedback.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing and translation of the manuscript, and to assist in writing and debugging experimental code. We have not used generative AI tools for generating research ideas, designing experiments, analyzing results, or producing any scientific claims, figures, or tables in this paper. We have reviewed all AI-assisted work: all AI-polished text was carefully checked by the authors for accuracy and faithfulness to our intended meaning, and all AI-assisted code was verified, tested, and debugged by the authors before being used in any experiment. We take full responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on algorithmic and methodological research on reinforcement learning for large language models. It does not involve human subjects, personally identifiable information, or newly collected data. All datasets and pre-trained models used in our experiments are publicly available and are used in compliance with their respective licenses and intended usage. We do not foresee any direct harmful societal impacts arising from this work beyond those generally associated with advances in language model capabilities.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure the reproducibility of our results. The model used in all experiments is the publicly available, and all training and evaluation datasets are public. Complete implementation details, including training configurations, hyperparameters, and evaluation protocols, are provided in Section 5.1 of the main text and Appendix D.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*, pp. 21246–21263, 2024.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. In *NeurIPS*, pp. 100213–100240, 2024.

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael R Lyu, and Irwin King. Binarybert: Pushing the limit of bert quantization. In *ACL-IJCNLP*, pp. 4334–4348, 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 *ACL*, pp. 10081–10100, 2025a.

Mengzhao Chen, Meng Wu, Hui Jin, Zhihang Yuan, Jing Liu, Chaoyi Zhang, Yunshui Li, Jie Huang, Jin Ma, Zeyue Xue, et al. Int vs fp: A comprehensive study of fine-grained low-bit quantization formats. *arXiv preprint arXiv:2510.25602*, 2025b.

Jatin Chhugani, Geonhwa Jeong, Bor-Yiing Su, Yunjie Pan, Hanmei Yang, Aayush Ankit, Jiecao Yu, Summer Deng, Yunqing Chen, Nadathur Satish, et al. Unveiling the potential of quantization with mxfp4: Strategies for quantization error reduction. *arXiv preprint arXiv:2603.08713*, 2026.

Musa Cim, Poovaiah Palangappa, Miro Hodak, Ravi Dwivedula, Meena Arunachalam, and Mahmut Taylan Kandemir. Pretraining large language models with mxfp4 on native fp4 hardware. *arXiv preprint arXiv:2605.09825*, 2026.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. *arXiv preprint arXiv:2605.00674*, 2026.

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 *ACL*, pp. 102–116, 2024.

Vage Egiazarian, Roberto Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, et al. Bridging the gap between promise and performance for microscaling fp4 quantization. In *ICLR*, pp. 113529–113563, 2026.

Shiyuan Feng, Huan-ang Gao, Haohan Chi, Hanlin Wu, Zhilong Zhang, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Weak-to-strong generalization via direct on-policy distillation. *arXiv preprint arXiv:2607.05394*, 2026.

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.

Team Glm, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *ICLR*, pp. 32694–32717, 2024.

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 *ICLR*, pp. 108059–108130, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

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

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *ICLR*, pp. 58791–58831, 2025.

Hyesung Jeon, Yulhwa Kim, and Jae-Joon Kim. L4q: parameter efficient quantization-aware fine-tuning on large language models. In *ACL*, pp. 2002–2024, 2025.

Minsoo Kim, Sihwa Lee, Suk-Jin Hong, Du-Seong Chang, and Jungwook Choi. Understanding and improving knowledge distillation for quantization aware training of large transformer encoders. In *EMNLP*, pp. 6713–6725, 2022.

Janghwan Lee, Sihwa Lee, Jinseok Kim, Yongjik Kim, Jieun Lim, Jinwook Oh, and Jungwook Choi. Reqat: Achieving full-precision reasoning accuracy with 4-bit floating-point quantization-aware training. *arXiv preprint arXiv:2606.15682*, 2026.

Ji-Fu Li, Manyi Zhang, Xiaobo Xia, Han Bao, Haoli Bai, Zhenhua Dong, and Xianzhi Yu. Batquant: Outlier-resilient mxfp4 quantization via learnable block-wise optimization. *arXiv preprint arXiv:2603.16590*, 2026a.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. In *NeurIPS*, pp. 87766–87800, 2024a.

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. In *MLSys*, pp. 87–100, 2024b.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *NeurIPS*, pp. 21558–21572, 2023.

Xiaohao Liu, Xiaobo Xia, Manyi Zhang, Ji-Fu Li, Xianzhi Yu, Fei Shen, Xiu Su, See-Kiong Ng, and Tat-Seng Chua. Freeact: Freeing activations for llm quantization. *arXiv preprint arXiv:2603.01776*, 2026a.

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 *ACL Findings*, pp. 467–484, 2024b.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *ICLR*, pp. 92009–92032, 2025.

Zechun Liu, Changsheng Zhao, Hanxian Huang, Sijia Chen, Jing Zhang, Jiawei Zhao, Scott Roy, Lisa Jin, Yunyang Xiong, Yangyang Shi, et al. Paretoq: Improving scaling laws in extremely low-bit llm quantization. In *NeurIPS*, pp. 91311–91336, 2026b.

Kevin Lu and Thinking Machines Lab. On-policy distillation. Thinking Machines Lab: Connectionism, 2025. URL https://thinkingmachines.ai/blog/on-policy-distillation/.

Keyu Lv, Manyi Zhang, Xiaobo Xia, Jingchen Ni, Shannan Yan, Xianzhi Yu, Lu Hou, Chun Yuan, and Haoli Bai. What makes low-bit quantization-aware training work for reasoning llms? a systematic study. *arXiv preprint arXiv:2601.14888*, 2026.

Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852*, 2017.

Open Compute Project. OCP microscaling formats (MX) specification, version 1.0. Technical report, Open Compute Project, 2023. URL https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*, 2018.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Bita Darvish Rouhani, Ritchie Zhao, Ankit More, Mathew Hall, Alireza Khodamoradi, Summer Deng, Dhruv Choudhary, Marius Cornea, Eric Dellinger, Kristof Denolf, et al. Microscaling data formats for deep learning. *arXiv preprint arXiv:2310.10537*, 2023.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *EuroSys*, pp. 1279–1297, 2025.

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

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Albert Tseng, Tao Yu, and Youngsuk Park. Training llms with mxfp4. *arXiv preprint arXiv:2502.20586*, 2025.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *NeurIPS*, pp. 95266–95290, 2024.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *ICML*, pp. 38087–38099, 2023.

Meng Xin, Sweta Priyadarshi, Jingyu Xin, Bilal Kartal, Aditya Vavre, Asma Kuriparambil Thekkumpate, Zijia Chen, Ameya Sunil Mahabaleshwarkar, Ido Shahaf, Akhiad Bercovich, et al. Quantization-aware distillation for nvfp4 inference accuracy recovery. *arXiv preprint arXiv:2601.20088*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

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.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026.

Manyi Zhang, Ji-Fu Li, Zhongao Sun, Haoli Bai, Hui-Ling Zhen, Zhenhua Dong, and Xianzhi Yu. Benchmarking post-training quantization of large language models under microscaling floating point formats. In *ACL*, pp. 39907–39923, 2026a.

Shihao Zhang, Haoyu Zhang, Ian Colbert, and Rayan Saab. Qronos: Correcting the past by shaping the future... in post-training quantization. In *ICLR*, pp. 87775–87798, 2026b.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

APPENDIX

A Detailed Derivations **15**

A.1 Assumption 15

A.2 Closed-Form Optimal Target Policy 15

A.3 Factorization of the Direction-Guided Target 16

A.4 Interpolation and Extrapolation 16

A.5 Exact Sequence Gradient and Zero-Discount Approximation 17

B Training Procedure **18**

C Additional Experimental Results **19**

C.1 Results without Macro Block Scaling 19

C.2 Reward-Scale Results 19

C.3 KL Regularization for Preference-Strengthening Deviations 19

D Supplementary Experimental Setup **20**

D.1 Model and Quantization Configuration 20

D.2 Training Configuration 20

D.3 Evaluation Configuration 20

D.4 Training Overhead 20

A Detailed Derivations

This appendix derives the idealized target policy in Section 4.2 and the on-policy update in Section 4.3. We first solve the sequence-level optimization problem exactly, and then derive its policy gradient and the zero-discount approximation used in practice.

A.1 Assumption

Fix a prompt x and a response $y = (y_1, \dots, y_L)$, with prefix $s_t = (x, y_{<t})$. We make the following assumption.

Assumption A.1 (Finite response space and common support)**.** *For every admissible prefix s_t, the teacher and quantized reference distributions share a fixed finite support $\mathcal{V}(s_t)$ and satisfy*

$$\pi_T(a \mid s_t) > 0, \qquad \pi_Q(a \mid s_t) > 0, \qquad \forall a \in \mathcal{V}(s_t).$$

There exists a finite $L_{\max}$ such that every admissible response terminates after at most $L_{\max}$ tokens. Moreover, π_T, π_Q, and the binary mask M remain fixed while optimizing π_θ.

Let $\mathcal{Y}(x)$ denote the finite set of complete responses induced by the admissible token sets, the termination rule, and the length bound $L_{\max}$. We identify each policy with its induced normalized distribution on $\mathcal{Y}(x)$. Assumption A.1 then implies $\pi_Q(y \mid x) > 0$ for every $y \in \mathcal{Y}(x)$.

Besides, define the sequence-level reward by

$$R_A(y \mid x) = \sum_{t=1}^{L} M(y_t \mid s_t) \log \frac{\pi_T(y_t \mid s_t)}{\pi_Q(y_t \mid s_t)}. \tag{13}$$

Because $\mathcal{Y}(x)$ is finite and all token-level quantities in Eq. (13) are finite, $R_A(\cdot \mid x)$ is a fixed, finite-valued function on $\mathcal{Y}(x)$. Consequently, the partition functions introduced below are finite, and all relevant log-ratios are well defined. Accordingly, the following analysis characterizes the distributional optimum over $\Delta(\mathcal{Y}(x))$, which serves as the ideal target for the parameterized W4A4 policy.

A.2 Closed-Form Optimal Target Policy

For a fixed prompt x, consider the unrestricted distributional counterpart of Eq. (8):

$$\max_{\pi(\cdot|x) \in \Delta(\mathcal{Y}(x))} \mathcal{F}_\lambda(\pi; x), \qquad \mathcal{F}_\lambda(\pi; x) = \lambda \mathbb{E}_{y \sim \pi(\cdot|x)}[R_A(y \mid x)] - D_{\mathrm{KL}}(\pi(\cdot \mid x) \| \pi_Q(\cdot \mid x)). \tag{14}$$

Proposition 1 (Ideal target policy)**.** *Under Assumption A.1, for any fixed finite λ, the objective $\mathcal{F}_\lambda(\pi; x)$ is strictly concave in $\pi(\cdot \mid x)$ and admits the unique maximizer*

$$\pi^*_\lambda(y \mid x) = \frac{\pi_Q(y \mid x) \exp(\lambda R_A(y \mid x))}{Z_\lambda(x)}, \qquad Z_\lambda(x) = \sum_{y' \in \mathcal{Y}(x)} \pi_Q(y' \mid x) \exp(\lambda R_A(y' \mid x)). \tag{15}$$

Proof. The reward term in Eq. (14) is linear in π. Since π_Q has full support, the negative KL term is strictly concave on the probability simplex. Hence $\mathcal{F}_\lambda$ is strictly concave.

To derive the maximizer, introduce a Lagrange multiplier η for the normalization constraint $\sum_y \pi(y \mid x) = 1$:

$$\mathcal{L}(\pi, \eta) = \sum_y \pi(y \mid x) \left[\lambda R_A(y \mid x) - \log \frac{\pi(y \mid x)}{\pi_Q(y \mid x)} \right] + \eta \left(\sum_y \pi(y \mid x) - 1 \right). \tag{16}$$

Differentiating with respect to $\pi(y \mid x)$ and setting the derivative to zero yields

$$\log \frac{\pi(y \mid x)}{\pi_Q(y \mid x)} = \lambda R_A(y \mid x) + \eta - 1. \tag{17}$$

Exponentiating both sides gives

$$\pi(y \mid x) = \pi_Q(y \mid x) \exp(\lambda R_A(y \mid x)) \exp(\eta - 1). \tag{18}$$

Imposing the normalization constraint $\sum_y \pi(y \mid x) = 1$ gives

$$1 = \exp(\eta - 1) \sum_{y \in \mathcal{Y}(x)} \pi_Q(y \mid x) \exp(\lambda R_A(y \mid x)) = \exp(\eta - 1) Z_\lambda(x). \tag{19}$$

Hence, $\exp(\eta - 1) = Z_\lambda(x)^{-1}$. Substituting this into Eq. (18) recovers Eq. (15).

Under Assumption A.1, $\mathcal{Y}(x)$ is finite, π_Q has full support, and R_A is finite-valued. Thus $0 < Z_\lambda(x) < \infty$, and the resulting distribution has full support and satisfies the stationarity condition. By strict concavity, it is the unique global maximizer. □

At unit reward scale, define $\pi_A(\cdot \mid x) = \pi_1^*(\cdot \mid x)$. Setting $\lambda = 1$ in Eq. (15) gives

$$\pi_A(y \mid x) = \frac{\pi_Q(y \mid x) \exp(R_A(y \mid x))}{Z_A(x)}, \qquad Z_A(x) = Z_1(x). \tag{20}$$

A.3 Factorization of the Direction-Guided Target

We next derive Eq. (9). For $s_t = (x, y_{<t})$, write $M_t = M(y_t \mid s_t)$. Using the autoregressive factorization of π_Q and Eq. (13),

$$\begin{aligned}
\pi_Q(y \mid x) \exp(R_A(y \mid x)) &= \prod_{t=1}^{L} \pi_Q(y_t \mid s_t) \exp\left(\sum_{t=1}^{L} M_t \log \frac{\pi_T(y_t \mid s_t)}{\pi_Q(y_t \mid s_t)} \right) \\
&= \prod_{t=1}^{L} \pi_Q(y_t \mid s_t) \left(\frac{\pi_T(y_t \mid s_t)}{\pi_Q(y_t \mid s_t)} \right)^{M_t} \\
&= \prod_{t=1}^{L} \pi_T(y_t \mid s_t)^{M_t} \pi_Q(y_t \mid s_t)^{1 - M_t}.
\end{aligned} \tag{21}$$

As M_t is evaluated at the prefix s_t visited by y, Eq. (21) assigns an unnormalized score to each complete response. Dividing these scores by $Z_A(x)$ yields the target distribution π_A over $\mathcal{Y}(x)$. The factorized form also recovers two familiar endpoints. Retaining every teacher correction gives $R_A(y \mid x) = \log[\pi_T(y \mid x)/\pi_Q(y \mid x)]$ and hence $\pi_A = \pi_T$, whereas withholding every correction gives $R_A(y \mid x) = 0$ and hence $\pi_A = \pi_Q$. Direction-guided correction generalizes these endpoints by retaining teacher corrections only where they follow the BF16 post-training direction.

A.4 Interpolation and Extrapolation

Eq. (15) characterizes the optimum at an arbitrary reward scale. To express it through the unit-scale target, first rearrange Eq. (20) as

$$\exp(R_A(y \mid x)) = Z_A(x) \frac{\pi_A(y \mid x)}{\pi_Q(y \mid x)}. \tag{22}$$

Raising both sides to the power λ and multiplying by $\pi_Q(y \mid x)$ gives

$$\pi_Q(y \mid x) \exp(\lambda R_A(y \mid x)) = Z_A(x)^\lambda \pi_A(y \mid x)^\lambda \pi_Q(y \mid x)^{1-\lambda}. \tag{23}$$

The factor $Z_A(x)^\lambda$ is independent of y and cancels under normalization. Therefore,

$$\pi_\lambda^*(y \mid x) = \frac{\pi_A(y \mid x)^\lambda \pi_Q(y \mid x)^{1-\lambda}}{\sum_{y'} \pi_A(y' \mid x)^\lambda \pi_Q(y' \mid x)^{1-\lambda}}, \tag{24}$$

which establishes Eq. (10) with its normalization made explicit.

The interpolation statement is most precisely expressed through response log-odds. For any two supported responses y and $\bar{y}$,

$$\log \frac{\pi^*_\lambda(y \mid x)}{\pi^*_\lambda(\bar{y} \mid x)} = \lambda \log \frac{\pi_A(y \mid x)}{\pi_A(\bar{y} \mid x)} + (1-\lambda) \log \frac{\pi_Q(y \mid x)}{\pi_Q(\bar{y} \mid x)}. \tag{25}$$

Hence $0 < \lambda < 1$ gives a log-linear interpolation between π_Q and π_A, $\lambda = 1$ recovers π_A, and $\lambda > 1$ extrapolates beyond π_A along the same log-odds direction. This relation characterizes the effect of λ on the idealized target distribution; its downstream effect is evaluated empirically in Section 5.3. When $M_t = 1$ everywhere, $\pi_A = \pi_T$ and Eq. (24) reduces to

$$\pi^*_\lambda(y \mid x) \propto \pi_T(y \mid x)^\lambda \pi_Q(y \mid x)^{1-\lambda}, \tag{26}$$

recovering the G-OPD target with π_Q as the reference policy. When the mask vanishes everywhere, Eq. (24) reduces to $\pi^*_\lambda = \pi_Q$ for every λ.

A.5 Exact Sequence Gradient and Zero-Discount Approximation

We finally derive the gradient used in Section 4.3, following the sequence-level score-function argument used for OPD. Define

$$c_{\theta,t}(y_t, s_t) = \lambda r_A(y_t \mid s_t) - \log \frac{\pi_\theta(y_t \mid s_t)}{\pi_Q(y_t \mid s_t)}. \tag{27}$$

By autoregressive factorization,

$$\lambda R_A(y \mid x) - \log \frac{\pi_\theta(y \mid x)}{\pi_Q(y \mid x)} = \sum_{t=1}^{L} c_{\theta,t}(y_t, s_t). \tag{28}$$

For compactness, write $C_\theta(y, x) = \sum_{t=1}^{L} c_{\theta,t}(y_t, s_t)$. Expanding the expectation in Eq. (8) gives

$$\begin{aligned}
\nabla_\theta \mathcal{J}_\lambda(\theta) &= \mathbb{E}_{x\sim\mathcal{D}} \nabla_\theta \sum_y \pi_\theta(y \mid x) C_\theta(y, x) \\
&= \mathbb{E}_{x\sim\mathcal{D}} \left[\sum_y \nabla_\theta \pi_\theta(y \mid x) C_\theta(y, x) \right] + \mathbb{E}_{x\sim\mathcal{D}} \left[\sum_y \pi_\theta(y \mid x) \nabla_\theta C_\theta(y, x) \right].
\end{aligned} \tag{29}$$

Only the student log-ratio in C_θ depends explicitly on θ. Consequently,

$$\begin{aligned}
\mathbb{E}_{x\sim\mathcal{D}} \left[\sum_y \pi_\theta(y \mid x) \nabla_\theta C_\theta(y, x) \right] &= -\mathbb{E}_{x\sim\mathcal{D}} \left[\sum_y \pi_\theta(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x) \right] \\
&= -\mathbb{E}_{x\sim\mathcal{D}} \left[\sum_y \nabla_\theta \pi_\theta(y \mid x) \right] = 0.
\end{aligned} \tag{30}$$

Using $\nabla_\theta \pi_\theta = \pi_\theta \nabla_\theta \log \pi_\theta$ in the remaining term yields the exact sequence-level gradient

$$\nabla_\theta \mathcal{J}_\lambda(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[C_\theta(y, x) \nabla_\theta \log \pi_\theta(y \mid x) \right]. \tag{31}$$

Expanding both factors autoregressively gives

$$\nabla_\theta \mathcal{J}_\lambda(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{L} \sum_{t'=1}^{L} c_{\theta,t'} \nabla_\theta \log \pi_\theta(y_t \mid s_t) \right]. \tag{32}$$

Terms with $t' < t$ vanish. Indeed, $c_{\theta,t'}$ is determined by $(x, y_{<t'})$ whenever $t' < t$, and therefore

$$\begin{aligned}
&\mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[c_{\theta,t'} \nabla_\theta \log \pi_\theta(y_t \mid s_t) \right] \\
&\quad = \mathbb{E}_{x\sim\mathcal{D},\, y_{<t}\sim\pi_\theta(\cdot|x)} \left[c_{\theta,t'} \mathbb{E}_{y_t\sim\pi_\theta(\cdot|s_t)} \left[\nabla_\theta \log \pi_\theta(y_t \mid s_t) \mid s_t \right] \right] \\
&\quad = \mathbb{E}_{x\sim\mathcal{D},\, y_{<t}\sim\pi_\theta(\cdot|x)} \left[c_{\theta,t'} \nabla_\theta \sum_{a\in\mathcal{V}(s_t)} \pi_\theta(a \mid s_t) \right] = 0.
\end{aligned} \tag{33}$$

Algorithm 1 Direction-guided quantization-aware on-policy distillation

Require: Prompt distribution $\mathcal{D}$; frozen post-trained BF16 teacher π_T; frozen BF16 base policy π_B; fixed W4A4 reference policy π_Q; reward scale $\lambda > 0$; learning rate η; number of iterations K
Ensure: Trained W4A4 policy π_θ

Initialize $\pi_\theta \leftarrow \pi_Q$
for $k = 1, \ldots, K$ **do**
  Sample a minibatch of prompts $\mathcal{B} \sim \mathcal{D}$
  for all $x_i \in \mathcal{B}$ **do**
    Sample $y_i = (y_{i,1}, \ldots, y_{i,L_i})$ from $\pi_\theta(\cdot \mid x_i)$
    for $t = 1, \ldots, L_i$ **do**
      Set $s_{i,t} \leftarrow (x_i, y_{i,<t})$
      $\Delta_{\mathrm{PT},i,t} \leftarrow \log \frac{\pi_T(y_{i,t}|s_{i,t})}{\pi_B(y_{i,t}|s_{i,t})}$, $\Delta_{\mathrm{Q},i,t} \leftarrow \log \frac{\pi_T(y_{i,t}|s_{i,t})}{\pi_Q(y_{i,t}|s_{i,t})}$
      $M_{i,t} \leftarrow \mathbf{1}[\Delta_{\mathrm{PT},i,t}\Delta_{\mathrm{Q},i,t} > 0]$
      $r_{A,i,t} \leftarrow M_{i,t}\Delta_{\mathrm{Q},i,t}$
      $\widehat{A}_{i,t} \leftarrow \mathrm{sg}[\lambda r_{A,i,t} - \log \frac{\pi_\theta(y_{i,t}|s_{i,t})}{\pi_Q(y_{i,t}|s_{i,t})}]$
    end for
  end for
  $\widehat{g} \leftarrow \frac{1}{|\mathcal{B}|} \sum_{x_i \in \mathcal{B}} \sum_{t=1}^{L_i} \widehat{A}_{i,t} \nabla_\theta \log \pi_\theta(y_{i,t} \mid s_{i,t})$
  Update $\theta \leftarrow \theta + \eta \widehat{g}$
end for
return π_θ

Removing these terms from Eq. (32) gives

$$\nabla_\theta \mathcal{J}_\lambda(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{L} \left(\sum_{t'=t}^{L} c_{\theta,t'} \right) \nabla_\theta \log \pi_\theta(y_t \mid s_t) \right]. \tag{34}$$

Thus, the exact coefficient for token y_t is the sum of the current and future token contributions. More generally, introduce the discounted return-to-go

$$G_t^{(\gamma)} = \sum_{t'=t}^{L} \gamma^{t'-t} c_{\theta,t'}, \qquad 0 \le \gamma \le 1. \tag{35}$$

Eq. (34) corresponds to $\gamma = 1$. The zero-discount approximation sets $\gamma = 0$ and therefore retains only the immediate contribution,

$$\sum_{t'=t}^{L} c_{\theta,t'} \approx c_{\theta,t}, \tag{36}$$

which leads to

$$\begin{aligned} \nabla_\theta \mathcal{J}_\lambda(\theta) &\approx \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{L} \mathrm{sg}[c_{\theta,t}]\, \nabla_\theta \log \pi_\theta(y_t \mid s_t) \right] \\ &= \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\sum_{t=1}^{L} \widehat{A}_t \nabla_\theta \log \pi_\theta(y_t \mid s_t) \right]. \end{aligned} \tag{37}$$

Here sg denotes the stop gradient operation. Substituting Eq. (27) recovers Eq. (11) and (12).

B Training Procedure

Algorithm 1 summarizes the optimization procedure. At each iteration, the current W4A4 policy generates the responses on which the three fixed policies are evaluated. All log-probabilities at position t refer to the same sampled token $y_{i,t}$ and prefix $s_{i,t}$. The directional mask acts on the teacher correction, while the reference-policy term remains active at every response position.

Table 3: **Results on Qwen3-8B without macro block scaling (MBS).** All models use MXFP4 W4A4 quantization. Scores are percentages. *Avg.* is the unweighted mean across seven benchmarks, and *Recovery* normalizes this mean by the BF16 result. The best quantized result in each column is shown in **bold**.

Type	Method	Knowledge	Code			Math		Science	Summary	
		MMLU-Pro	LCB v6	MBPP+	HE+	AIME24	AIME25	GPQA	Avg.	Recovery
	BF16	73.77	43.27	74.20	83.50	71.46	63.12	58.13	66.78	100.00
PTQ	RTN	68.13	33.63	67.72	76.52	53.33	47.08	49.80	56.60	84.76
QAT	QAD	68.46	36.42	69.91	80.03	57.71	50.00	51.06	59.08	88.48
	OPD	**70.50**	37.56	69.78	79.57	56.67	50.21	**53.18**	59.64	89.31
	G-OPD	68.24	37.54	68.20	**82.01**	59.58	48.13	51.97	59.38	88.92
	Direct-OPD	68.09	36.35	65.15	76.52	58.33	51.25	52.47	58.31	87.32
	DQ-OPD	69.46	**39.19**	**70.83**	79.88	**63.96**	**57.08**	51.87	**61.75**	**92.47**

Table 4: **Results on Qwen3-30B-A3B without macro block scaling (MBS).** All models use MXFP4 W4A4 quantization. Scores are percentages. *Avg.* is the unweighted mean across seven benchmarks, and *Recovery* normalizes this mean by the BF16 result. The best quantized result in each column is shown in **bold**.

Type	Method	Knowledge	Code			Math		Science	Summary	
		MMLU-Pro	LCB v6	MBPP+	HE+	AIME24	AIME25	GPQA	Avg.	Recovery
	BF16	77.48	48.10	76.65	85.98	75.00	69.79	62.98	70.85	100.00
PTQ	RTN	73.71	42.18	71.69	80.49	68.54	56.67	57.93	64.46	90.97
QAT	QAD	73.80	44.53	72.42	82.47	69.79	58.13	60.10	65.89	93.00
	OPD	73.01	44.72	74.60	83.99	71.67	62.92	59.60	67.22	94.86
	G-OPD	74.24	44.76	**75.20**	**86.43**	70.63	61.88	59.60	67.53	95.31
	Direct-OPD	71.48	43.63	71.23	83.08	**73.96**	63.96	56.82	66.31	93.58
	DQ-OPD	**74.93**	**45.97**	74.60	84.45	70.83	**67.92**	**62.98**	**68.81**	**97.12**

C Additional Experimental Results

C.1 Results without Macro Block Scaling

Tables 3 and 4 report results under MXFP4 W4A4 quantization without macro block scaling (MBS).

C.2 Reward-Scale Results

Table 5 provides the full benchmark results for the reward-scale ablation shown in Figure 5c. We vary λ while keeping the direction-guided correction criterion and the KL regularization toward the quantized reference policy π_Q unchanged. Results are reported separately for Qwen3-8B and Qwen3-30B-A3B across all evaluation benchmarks.

C.3 KL Regularization for Preference-Strengthening Deviations

In Section 5.3, we investigated whether KL regularization toward the quantized reference policy π_Q is necessary for tokens identified as preference-strengthening deviations. Table 6 provides the detailed performance breakdown across all seven benchmarks for both Qwen3-8B and Qwen3-30B-A3B. The results confirm that retaining the KL term consistently prevents performance degradation, particularly in complex reasoning and coding tasks.

Table 5: **Reward-scale ablation.** We vary the reward scale λ while keeping the direction-guided correction criterion and the KL regularization toward the quantized reference policy π_Q unchanged.

Model	λ	Knowledge	Code			Math		Science	Summary
		MMLU-Pro	LCB v6	MBPP+	HE+	AIME24	AIME25	GPQA	Avg.
Qwen3-8B	1	70.15	38.53	71.20	82.77	68.96	59.38	54.24	63.60
	1.25	69.82	37.77	73.08	82.77	70.83	60.63	56.36	64.47
	2	68.82	38.72	70.80	80.80	67.71	61.46	55.56	63.40
	4	67.05	37.54	69.70	83.70	72.29	65.83	53.03	64.16
Qwen3-30B-A3B	1	74.09	43.93	75.13	86.13	72.92	64.58	61.36	68.31
	1.25	75.15	43.77	75.93	85.77	72.92	66.88	61.92	68.90
	2	74.85	43.91	76.06	85.37	74.58	67.71	61.57	69.15
	4	75.87	46.28	76.32	85.67	75.42	68.96	62.47	70.14

Table 6: **Ablation study on the KL regularization for preference-strengthening deviations.** We compare the full DQ-OPD (w/ KL) against a variant that selectively removes the KL regularization for preference-strengthening tokens (w/o KL). Results are reported as percentages.

Model	Setting	Knowledge	Code			Math		Science	Summary
		MMLU-Pro	LCB v6	MBPP+	HE+	AIME24	AIME25	GPQA	Avg.
Qwen3-8B	w/ KL	69.82	37.77	**73.08**	82.77	**70.83**	**60.63**	**56.36**	**64.47**
	w/o KL	67.74	37.49	67.13	80.18	67.29	58.54	53.94	61.76
Qwen3-30B-A3B	w/ KL	75.87	**46.28**	76.32	**85.67**	**75.42**	**68.96**	62.47	**70.14**
	w/o KL	**76.09**	42.80	**76.92**	84.91	73.54	63.33	**63.43**	68.72

D Supplementary Experimental Setup

We provide the complete experimental configuration used throughout the paper. Tables 7, 8, and 9 summarize the model and quantization setup, training configuration, and evaluation protocol, respectively.

D.1 Model and Quantization Configuration

We evaluate Qwen3-8B and Qwen3-30B-A3B (Yang et al., 2025), representing dense and Mixture-of-Experts (MoE) architectures. For each model, π_B denotes the BF16 base policy, π_T the post-trained BF16 teacher, and π_Q the MXFP4-quantized version of π_T, which serves as both the student initialization and the quantized reference policy.

D.2 Training Configuration

All methods are implemented using verl, with FSDP2 for distributed training and vLLM for rollout generation. Training prompts are sampled from OpenThoughts3-1.2M. QAD uses offline responses, whereas OPD-based methods generate fresh responses from the current student policy.

D.3 Evaluation Configuration

We evaluate seven benchmarks spanning general knowledge, code generation, mathematical reasoning, and scientific reasoning. All evaluations are conducted with lighteval. Unless otherwise specified, generation uses temperature 0.6 and top-p 0.95.

D.4 Training Overhead

During training, DQ-OPD requires token-level scores from π_T, π_B, and π_Q. The direction-guided correction itself introduces only lightweight element-wise operations once these scores are available.

Table 7: **Model and quantization configuration.**

Item	Setting
Models	Qwen3-8B; Qwen3-30B-A3B
Model architecture	Dense Transformer; Mixture-of-Experts Transformer
BF16 base policy	π_B
BF16 teacher	Post-trained policy π_T
Quantized policy	π_Q, obtained by quantizing π_T to MXFP4 W4A4
Student initialization	π_Q
Reference policy	π_Q
Weight precision	MXFP4
Activation precision	MXFP4
Quantized modules	Q/K/V/output projections in self-attention; gate/up/down projections in dense MLPs and MoE experts

Table 8: **Training configuration.**

Item	Setting
Training data	OpenThoughts3-1.2M
Batch size	256 prompts per update
Response source	Offline responses for QAD; 1 rollout response per prompt for OPD-based methods
Maximum prompt length	2,048 tokens
Maximum response length	8,192 tokens
Learning rate	1×10^{-5}
Warmup	None
Training steps	200
Rollout temperature	1.0
Rollout top-p	1.0
Reward scale λ	1.25 for Qwen3-8B; 4 for Qwen3-30B-A3B
Training backend	FSDP2
Rollout backend	vLLM

Table 9: **Evaluation configuration.**

Item	Setting
Evaluation framework	LIGHTEVAL
Datasets	MMLU-Pro, LiveCodeBench v6, MBPP+, HumanEval+, AIME 2024, AIME 2025, GPQA Diamond
Evaluation temperature	0.6
Evaluation top-p	0.95
Metrics	AIME: avg@16 over 16 responses per problem; GPQA Diamond: pass@1 with 10 responses per problem; Code benchmarks: pass@1 with 4 responses per problem; MMLU-Pro: pass@1 with 1 response per problem

At inference time, only the trained W4A4 student is used, with no additional teacher or reference model.