

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 TOWARDS QUANTIZATION-AWARE TRAINING FOR ULTRA-LOW-BIT REASONING LLMS

Anonymous authors

Paper under double-blind review

ABSTRACT

Large language models (LLMs) have achieved remarkable performance across diverse reasoning tasks, yet their deployment is hindered by prohibitive computational and memory costs. Quantization-aware training (QAT) enables ultra-low-bit compression (< 4 bits per weight), but existing QAT methods often degrade reasoning capability, partly because complex knowledge structures are introduced during the post-training process in LLMs. In this paper, through a systematic investigation of how quantization affects different data domains, we find that its impact on pre-training and reasoning capabilities differs. Building on this insight, we propose a novel two-stage QAT pipeline specifically designed for reasoning LLMs. In the first stage, we quantize the model using mixed-domain calibration data to preserve essential capabilities across domains; in the second stage, we fine-tune the quantized model with a teacher-guided reward-rectification loss to restore reasoning capability. We first demonstrate that mixed-domain calibration outperforms single-domain calibration at maximum 2.74% improvement on average over six tasks including reasoning and pre-trained tasks. Following experiments on five reasoning benchmarks show that our 2-bit-quantized Qwen3-8B outperforms post-training quantization (PTQ) baselines by 50.45% on average. Moreover, compared to ultra-low-bit-specialized models such as BitNet-2B4T, our pipeline achieves about 2% higher mathematical-reasoning accuracy using only [968M training tokens](#).

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable performance across various tasks, including mathematics (Shao et al., 2024; Wang et al.; Yang et al., 2024), coding (Hui et al., 2024; Roziere et al., 2023), and knowledge-intensive question answering (Lu et al., 2022). However, their prohibitive computational and memory requirements pose significant challenges for deployment in inference. One promising direction for reducing these inference costs is weight quantization (Zhou et al., 2024; Lang et al., 2024), which employs low-bit widths for model weights. Among various quantization methods, *quantization-aware training* (QAT), which fine-tunes the model with quantized weights, is especially effective for *ultra-low-bit widths* (< 4 bits) (Wang et al., 2023; Ma et al., 2024; Xu et al., 2024), enabling us to deploy lightweight and fast LLMs. For example, 2-bit quantized LLMs via QAT can achieve performance comparable to their pre-quantized fp16 counterparts (Ma et al., 2024; Kaushal et al., 2024; Liu et al., 2025c).

Despite the promising performance of QAT, existing approaches suffer from severe performance degradation on reasoning benchmarks (Du et al., 2024), such as mathematics, and instruction-following tasks (Lee et al., 2025). We hypothesize that this degradation arises from the complex knowledge structures introduced during post-training. The post-training process is an extensive process that includes supervised fine-tuning (Wei et al., 2021) and preference optimization (Ouyang et al., 2022; Rafailov et al., 2023), introducing new reasoning capabilities with existing commonsense knowledge acquired during pre-training. While it creates heterogeneous knowledge structures, it remains unclear how quantization affects the model’s performance on reasoning capabilities and pre-trained commonsense knowledge.

To address this gap, we conduct a systematic investigation of how quantization impacts different knowledge domains in post-training LLMs. Our analysis reveals that quantization creates inherent trade-offs between commonsense knowledge preservation and reasoning capability retention, where different domains exhibit varying sensitivity to quantization. Specifically, while performance on

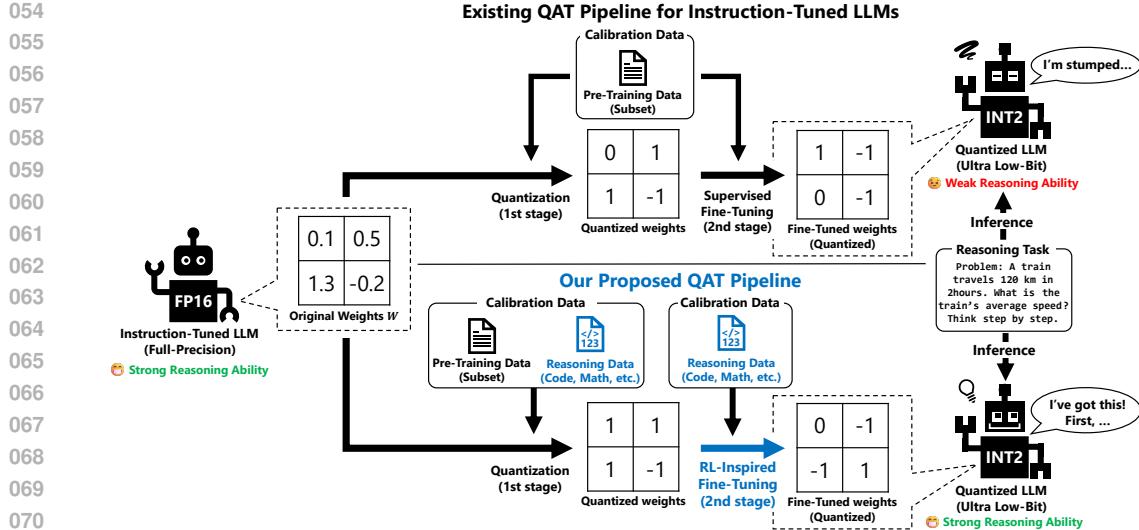


Figure 1: Comparison of the existing QAT pipeline with the proposed pipeline.

commonsense knowledge benchmarks remains relatively stable even with quantization using out-of-domain data, reasoning capabilities exhibit significant sensitivity to quantization data, suggesting that different domains have distinct requirements for effective quantization.

Based on this analysis, we introduce a quantization framework specifically designed for post-trained LLMs that address diverse knowledge domains through a novel two-stage pipeline. Following our observation, our quantization framework is designed to dedicate computational resources to maintain reasoning capability, with minimal efforts to preserve general knowledge. Specifically, the first stage carries out block-wise quantization with mixed-domain calibration. This mixed-domain calibration preserves essential reasoning capabilities that are difficult to restore, while also maintaining commonsense knowledge. Subsequently, we perform end-to-end fine-tuning with reinforcement learning inspired objectives to enhance reasoning capability. This unified framework enables extremely low-bit quantization of post-trained LLMs with minimal reasoning performance degradation.

Extensive experiments on five reasoning benchmarks demonstrate the effectiveness of our approach. Our method achieves significant improvements over existing post-training quantization methods for reasoning LLMs. Specifically, our 2-bit quantized Qwen3-8B outperforms other quantization methods by 50.45% on average. Notably, even when compared to specialized ternary LLMs like BitNet-2B4T, our 2-bit model with 1.7B parameters demonstrates superior mathematical reasoning performance with substantially reduced training costs—achieving 2.5% improvement using only 40K training sequences.

Our contributions can be summarized as follows:

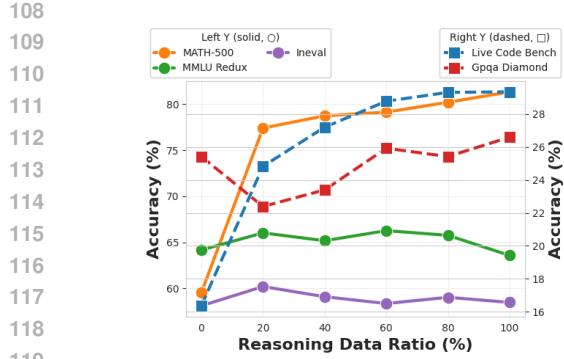
- We empirically demonstrate how quantization differently affects commonsense knowledge acquired during pre-training and reasoning capabilities developed in post-training. Our results highlight the importance of designing mixed calibration data to effectively preserve both of them.
- We propose two-stage quantization pipeline for post-trained LLMs that combines mixed-domain calibration and RL-inspired fine-tuning to preserve reasoning capabilities while achieving extremely low-bit quantization.
- We demonstrate that our approach achieves state-of-the-art (SOTA) performance on multiple reasoning benchmarks with both 2-bit and 3-bit quantization.

2 PRELIMINARIES

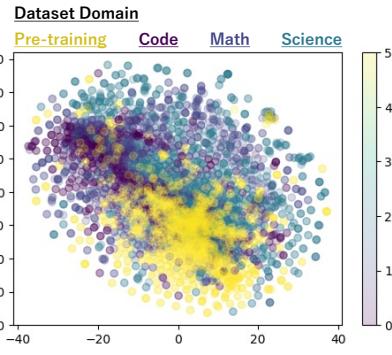
This section outlines the weight quantization and recent quantization-aware training (QAT) pipeline.

Weight Quantization: Weight quantization maps the model weights to low-bit width counterparts. Given a full-precision weight $w \in \mathbb{R}$, we obtain its dequantized approximation w' via

$$q := \text{clamp} \left(\lfloor w/s \rfloor + z, 0, 2^N - 1 \right), \quad w' := s(q - z),$$



(a) Performance for general (green ● and purple ●) and mathematical (orange ●, blue ■, and red ■) tasks on the reasoning data ratio.



(b) t-SNE visualization of the activations in the 14th Transformer block, where colors denote dataset domains.

Figure 2: Impact of data-domain composition on Qwen3-1.7B. In the ultra-low-bit quantized model, replacing a part of the calibration data drawn from the pre-training data (FineWeb-Edu) with reasoning data (OpenThoughts3-1.2M) leaves general-task accuracy unchanged while improving reasoning accuracy (left). In the full-precision model, t-SNE of the model’s activations shows tight clusters for pre-training inputs but wide dispersion for reasoning inputs (right).

where $s > 0$ is the scale factor, $z \in \mathbb{R}$ is the zero point, and N is the target bit width. $\lfloor \cdot \rfloor$ represents the nearest integer function (i.e., the round function), and $\text{clamp}(x, a, b)$ clamps input x to the interval $[a, b]$. Since the scale s and zero point z are shared across groups of weights (e.g., an entire matrix, a channel, or a block), each weight is represented only by an N -bit code q , with s and z stored once per group, achieving a low-bit width per-weight representation.

Weight quantization approaches are mainly categorized into two strategies: 1) post-training quantization (PTQ), which converts pre-trained model weights to low-bit widths without retraining; and 2) quantization aware training (QAT), which quantizes and fine-tunes the weights simultaneously. PTQ can quickly quantize weights with small or even without calibration data, while it struggles with ultra-low-bit quantization. In contrast, QAT can flexibly fine-tune the full-precision weights w , scaling factor s , and zero point z , achieving performance comparable to the full-precision model in ultra-low-bit scenarios.

Quantization-Aware Training Pipeline: As illustrated in the top panel of Figure 1, existing QAT pipelines generally comprise two stages: 1) an initial quantization stage; and 2) a fine-tuning stage. The first stage initializes the quantized weights that serve as the starting point for the subsequent stage. Some methods omit this step, whereas the latest state-of-the-art (SOTA) QAT approaches (Chen et al., 2024; Du et al., 2024) have demonstrated that quantizing weights using a subset of the pre-training dataset as calibration data enables stable fine-tuning during the subsequent stage. In the second stage, the weights quantized in the first stage are fine-tuned by minimizing a training objective. All parameters (i.e., weights, scale, zero point) or some of them are fine-tuned, and this paper fine-tunes only the scale, following one of the SOTA QAT approaches, EfficientQAT. Also, the training objective is typically either a self-supervised pre-training loss (Liu et al., 2025c; Chen et al., 2024) or a knowledge-distillation loss (Du et al., 2024; Lee et al., 2025).

3 REASONING-ORIENTED TWO-STAGE QUANTIZATION AWARE TRAINING

This section proposes a novel QAT pipeline that enables the preservation of reasoning capabilities after ultra-low-bit quantization. We first analyze the impact of quantization on various knowledge domains, and based on these findings, we introduce the reasoning-oriented QAT pipeline.

3.1 QUANTIZATION IMPACTS ACROSS KNOWLEDGE DOMAINS

This section analyzes how domain selection for calibration data affects the overall model performance. Existing quantization approaches mainly perform quantization with either pre-training

162 data (Liu et al., 2025c; Chen et al., 2024), or domain-specific data, such as mathematics (Liu et al.,
 163 2025a). While previous work has selected calibration data tailored to specific target tasks, the cross-
 164 task implications of such task-specific calibration choices remain largely unexplored.

165 To investigate the effect of domain selection for calibration data, we analyze the impact of selecting
 166 different calibration datasets on performance across multiple tasks and knowledge domains. As
 167 shown in the results of Qwen3-1.7B quantized to 3-bits by EfficientQAT (Chen et al., 2024) (Figure
 168 2a), the tasks can be broadly grouped into two trends: 1) tasks for which performance improves
 169 as the amount of reasoning data increases; and 2) tasks for which performance remains almost
 170 constant regardless of the amount of reasoning data. Notably, all tasks in the first category are rep-
 171 resented in the reasoning dataset, i.e., code (blue), mathematical tasks (orange), scientific questions
 172 and answers (red). These results indicate that reasoning data tends to suffer from domain shift, while
 173 tasks related to commonsense knowledge are less sensitive to calibration datasets. On the other hand,
 174 common tasks also demonstrate performance degradation when calibrated with pure reasoning data,
 175 suggesting the importance of dataset diversity even for tasks that appear less sensitive to calibration
 176 choices.

177 These distinct trends happen as the intermediate distributions between pre-trained data and reasoning
 178 data differ, as shown in Figure 2b. The distributional mismatch leads to suboptimal quantization
 179 performance when calibrating on single-domain data, resulting in higher quantization errors for
 180 tasks that require domain-specific representations.

181 3.2 PROPOSED METHOD

182 We now introduce the novel QAT pipeline for ultra-low-bit reasoning LLMs.

183 **Knowledge Domain Selection in Calibration data:** We first focus on the mixing ratio of the
 184 knowledge domain in calibration data. The results in Section 3.1 illustrate the importance of select-
 185 ing appropriate calibration data when a QAT pipeline is applied to post-trained LLMs. In particular,
 186 it is important to mix pre-training data and reasoning data in an appropriate ratio. Building on these
 187 findings, we propose using novel calibration data in the first stage of the QAT pipeline. This data is
 188 composed of 80% reasoning-focused data and 20% pre-training data, designed to bias the calibration
 189 process toward reasoning while retaining coverage of pre-training distributions.

190 **Supervised Fine-Tuning With Reward Rectification Loss:** We secondly aim at the fine-tuning
 191 stage. Quantization of the first stage using proposed calibration data mixed with pre-training data
 192 preserves the fundamental capabilities of the LLM, enabling us to focus on enhancing reasoning
 193 capabilities during the fine-tuning stage. A straightforward approach to enhance reasoning ability
 194 is to perform supervised fine-tuning using reasoning data, but such training does not effectively
 195 generalize into reasoning data (Chu et al., 2025). Employing reinforcement learning could improve
 196 generalization on reasoning data, but online text generation incurs auto-regressive text generations,
 197 resulting in huge training overhead. To balance training efficiency and generalization on unseen
 198 data, we employ reweighted rectification (Wu et al., 2025) for supervised fine-tuning to make the
 199 objective function reinforcement-like.

200 Reward rectification is scaling factors for the loss function in supervised fine-tuning. Given the
 201 datasets $\mathcal{D} = \{x, y^*\}$ and the supervised fine-tuning loss $\mathcal{L}_{SFT}(\theta)$, reward rectification loss $\mathcal{L}(\theta)$
 202 dynamically reweights the supervised loss as follows:

$$203 \mathcal{L}(\theta) = \mathcal{L}_{SFT}(\theta) \cdot \text{sg}(1/w),$$

204 where w is the dynamic reweighting factor and $\text{sg}(\cdot)$ denotes the stop-gradient operator.

205 This formulation can be viewed as bridging supervised fine-tuning and reinforcement learning. In
 206 particular, choosing $w = 1/\pi_\theta(y | x)$ yields a gradient equivalent to an on-policy policy-gradient
 207 update with the reward function:

$$208 r(x, y) = \mathbf{1}[y = y^*],$$

209 where $\pi_\theta(y | x)$ is the model’s conditional probability of generating an output y given an input x
 210 under parameters θ . This dynamic re-weighting can avoid over-concentration on low-probability
 211 reference tokens, improving generalization despite not using additional sampling or reward func-
 212 tions.

216
 217 Table 1: Accuracy comparison for different calibration data on 6 benchmarks. Higher values are
 218 better. We define the group size as 128. Mixed data contains 80% of reasoning data and 20% of
 219 pre-training data.

220 221 222	223 224 225	226 227	228 229 230	231 232	233 234 235	Reasoning Tasks		Pre-trained Tasks		236 237
						Model (Qwen3)	Bit Width	Dataset Type	MATH-500	
238 239 240 241 242 243	1.7B	Pre-training	59.53	16.36	25.42	64.16	57.96	58.60	47.01	
		Reasoning	81.33	29.35	26.60	63.58	55.82	59.52	52.70	
		Mixed	80.20	29.32	25.42	65.76	56.71	59.70	52.85	
	4B	Pre-training	0.13	0.00	0.00	0.00	50.75	13.86	10.79	
		Reasoning	20.60	0.09	7.58	19.93	44.77	25.51	19.75	
		Mixed	18.68	0.47	7.58	28.92	49.11	24.58	21.56	
244 245 246 247 248 249	8B	Pre-training	81.80	35.42	41.92	77.98	63.94	71.16	62.04	
		Reasoning	90.90	46.89	42.76	78.55	63.71	71.72	65.76	
		Mixed	90.70	46.85	45.45	78.91	63.97	74.86	66.79	
	16B	Pre-training	2.73	0.00	6.23	26.89	59.34	17.74	18.82	
		Reasoning	33.80	5.78	11.62	46.26	53.41	32.90	30.63	
		Mixed	22.60	5.12	14.14	51.20	55.89	31.05	30.00	
250 251 252 253 254 255	16B	Pre-training	87.00	37.25	41.66	82.60	69.19	76.34	65.67	
		Reasoning	91.80	53.84	51.52	81.75	68.14	75.79	70.47	
		Mixed	92.40	51.75	48.99	83.33	69.16	81.15	71.13	
	32B	Pre-training	5.33	0.28	4.55	41.72	63.35	19.41	22.44	
		Reasoning	42.27	8.15	10.44	51.40	55.05	35.86	33.86	
		Mixed	40.00	7.30	11.11	57.11	60.81	43.25	36.60	

While the original reward rectification uses the student model’s own probability $\pi_\theta(y \mid x)$ for reweighting, in the QAT, the quantized model’s distribution becomes less reliable due to precision loss. Using the quantized model’s own probabilities for reweighting could amplify these errors.

Therefore, we leverage the teacher model’s probability $\pi_t(y^*|x)$ as a more reliable reference for the reweighting factor. This teacher-guided approach ensures that the reweighting process is based on the target distribution we aim to recover, rather than the potentially corrupted distribution of the quantized model.

Thus, we introduce teacher-guided reward rectification loss $\mathcal{L}(\theta)$, where the teacher model π_t controls the scale of supervised loss function. Given the teacher probability with labeled data $\pi_t(y^* \mid x)$, teacher guided reward rectification loss can be represented as:

$$\mathcal{L}_t(\theta) = \mathcal{L}_{\text{SFT}}(\theta) \cdot \text{sg}(\pi_t(y^*|x)).$$

Intuitively, this formulation represents that the supervised loss values are amplified when the probability of the quantized model for the label is smaller than that of the teacher probability. When the distribution of the quantized model becomes close to the original distribution, this scaling factor acts as the original reward rectification.

To align the overall probabilistic distribution of the quantized model with original LLMs, we further introduce an additional KL divergence loss. Finally, our training loss function can be represented as:

$$\mathcal{L}(\theta) = \alpha \mathcal{L}_t(\theta) + \beta D_{\text{KL}}(\pi_t(\cdot|x) \parallel \pi_S(\cdot|x)), \quad (1)$$

where $D_{\text{KL}}(\pi_t(\cdot|x) \parallel \pi_S(\cdot|x)) = \sum_y \pi_t(y|x) \log \frac{\pi_t(y|x)}{\pi_S(y|x)}$ is the KL divergence between the fp16 model and the quantized model, and α, β is hyperparameters that control the effects of teacher-guided reward rectification loss and kl divergence loss.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Training: We conduct experiments on Qwen3 instruction-tuned models (Yang et al., 2025). For block-wise calibration, we use a total 4,096 samples with a context length of 2,048. Calibration

270 datasets consist of 80% of sequences sampled from OpenThoughts-1.2M (Guha et al., 2025) and the
 271 remaining 20% sampled from FineWeb-Edu (Lozhkov et al., 2024). We use different learning rates
 272 for quantization parameters (1e-4) and weight parameters (1e-5). For 2-bit quantization, we use a
 273 larger learning rate of 2e-5 for weights.

274 During supervised fine-tuning, models are with 32,768 samples from OpenThoughts-1.2M. We
 275 optimize all trainable parameters with the same learning rate. The learning rate for 3-bit quantization
 276 is 1e-6, while we use a larger learning rate for 2-bit quantization, 5e-6 for the 1.7B parameter, and
 277 1e-4 for other parameters. We use the AdamW optimizers (Loshchilov & Hutter, 2019) with the
 278 cosine annealing learning rate decay (Loshchilov & Hutter, 2017). Models are fine-tuned with a
 279 batch size of 64 and one epoch for 3-bit models, [except for Qwen 1.7B at 2-bit which uses 3 epochs](#).
 280 We filter out the top-20 probabilities for the KL loss. We set $\alpha = 0.2$ and $\beta = 1.0$ in Equation (1)
 281 unless explicitly stated otherwise.

282 **Evaluation:** We evaluate the zero-shot accuracy on five benchmarks including We evaluate the
 283 zero-shot accuracy on five benchmarks, including MATH-500 (Lightman et al., 2023), Live Code
 284 Bench (White et al., 2024), MMLU-Redux (Gema et al., 2024), GPQA- Diamond (Rein et al.,
 285 2024), and IFEval (Zhou et al., 2023), using the evalscope (Team, 2024). These tasks are evaluated
 286 in open-ended text generation. We use token-level sampling, whose tokens are sampled from the
 287 top 20 highest tokens with a temperature of 0.6. We basically use a maximum sequence length of
 288 32K for all benchmarks. However, we reduce the maximum sequence length to 8K on the lower-
 289 performance models to avoid excessive text generation due to the absence of a stop token. All
 290 evaluations are conducted three times, and we report the average accuracy.

291 **Quantization baselines to post-trained LLMs:** We compare our method with two PTQ quantiza-
 292 tion baselines, GPTQ (Frantar et al., 2022) and AWQ (Lin et al., 2024), both of which are evaluated
 293 on reasoning benchmarks (Liu et al., 2025a). To quantize these two baselines, we follow a similar
 294 strategy as conducted by Liu et al. (2025a). Specifically, we perform quantization using 128 sam-
 295 ples from NuminaMath (LI et al., 2024). We reproduce these quantized models locally, except for
 296 3 and 4 bits AWQ quantization for Qwen-8B, as reproduced performance is much inferior to the
 297 performance claimed in the paper (Liu et al., 2025a).

299 4.2 DATASET EFFECTS ON OVERALL PERFORMANCE

300 This section evaluates our proposed calibration datasets against single-domain calibrations using
 301 either pre-training data or reasoning data. We evaluate five benchmarks described in Section 4.1
 302 with the additional commonsense reasoning tasks (CSR) to evaluate general knowledge of quan-
 303 tized models. The CSR includes five subset tasks: ARC-e, ARC-c (Clark et al., 2018), PIQA (Bisk
 304 et al., 2020), HellaSwag (Zellers et al., 2019), and Winogrande Sakaguchi et al. (2021). Table 1
 305 demonstrates that mixed domain calibration outperforms single-domain calibration across various
 306 parameters in both 2-bit and 3-bit quantization settings. Compared with pre-training datasets, its
 307 performance improvements on reasoning benchmarks, including mathematical and coding tasks, are
 308 particularly notable. In addition, mixed data achieves comparable performance to pre-trained data
 309 and other benchmarks in reasoning tasks. When compared with reasoning-only datasets, the per-
 310 formance of mixed datasets on reasoning benchmarks is close, while performance in pre-trained
 311 domain benchmarks tends to be superior. These results suggest that including a small portion of
 312 pre-training data can be effective in maintaining commonsense knowledge. resulting in better gen-
 313 eralization results across a wide range of tasks.

315 4.3 COMPARISON WITH PRIOR QUANTIZATION APPROACHES FOR REASONING

316 This section compares our proposal with other quantization methods, including GPTQ and AWQ.
 317 Table 2 shows that our proposal significantly outperforms existing quantization approaches in both
 318 2-bit and 3-bit quantization settings. The performance improvements are particularly notable in 2-
 319 bit quantization settings. While the performance of GPTQ or AWQ quantized models is extremely
 320 low, our quantized models not only achieve solid performance but also exhibit steadily increasing
 321 accuracy as the number of parameters grows. This trend suggests that our quantization method
 322 scales effectively with model size even for extremely low-bit quantization. For 3-bit quantization,
 323 our approach dramatically improves performance, particularly on smaller models. For example, our

324 Table 2: Comparison of quantization methods and bit-widths on Qwen3 models. Values are %
 325 (higher is better). We define the group size as 128.

Model	Method	Bit Width(W/A)	Benchmarks (%)					Avg.
			MATH500	LiveCodeBench	MMLU-Redux	GPQA-Diamond	IFEval	
Qwen3-1.7B								
	FP (Baseline)	bfloat16	89.0	53.6	74.7	38.4	70.4	65.2
	GPTQ	4/16	86.3	36.9	70.2	34.3	66.2	58.8
	AWQ	4/16	87.4	44.4	71.6	35.9	65.3	60.9
	GPTQ	3/16	58.2	0.0	42.4	9.3	31.8	28.3
	AWQ	3/16	58.0	5.8	53.5	17.5	47.7	36.5
	Proposal	3/16	82.7	33.0	67.7	31.7	61.0	55.2
	GPTQ	2/16	2.1	0.0	5.8	4.7	8.5	4.2
	AWQ	2/16	0.0	0.0	27.3	8.1	12.3	9.5
	Proposal	2/16	48.6	6.5	40.1	14.5	32.2	28.4
Qwen3-4B								
	FP (Baseline)	bfloat16	93.6	71.2	84.3	51.5	83.6	76.8
	GPTQ	4/16	93.4	66.2	82.1	50.7	81.2	74.7
	AWQ	4/16	93.0	65.7	83.0	50.2	81.0	75.6
	GPTQ	3/16	85.0	21.2	65.6	24.9	54.7	50.3
	AWQ	3/16	88.3	37.2	74.2	33.7	71.9	61.1
	Proposal	3/16	89.5	50.2	79.8	46.8	75.3	68.3
	GPTQ	2/16	3.7	0.0	7.5	8.9	8.5	5.7
	AWQ	2/16	0.0	0.0	0.0	0.0	11.8	2.4
	Proposal	2/16	77.1	19.5	61.7	26.9	50.1	47.1
Qwen3-8B								
	FP (Baseline)	bfloat16	94.0	73.0	87.3	61.6	86.5	80.5
	GPTQ	4/16	94.6	69.6	86.8	58.1	86.9	79.2
	AWQ	4/16	97.0	54.7	N/A	59.6	N/A	N/A
	GPTQ	3/16	92.1	39.4	79.0	46.8	76.9	66.8
	AWQ	3/16	92.9	35.3	N/A	46.8	N/A	N/A
	Proposal	3/16	91.5	60.0	84.5	47.5	78.8	72.5
	GPTQ	2/16	2.8	0.0	6.4	5.2	8.7	4.6
	AWQ	2/16	0.0	0.0	5.9	3.9	10.2	4.0
	Proposal	2/16	80.4	28.5	72.6	34.5	59.3	55.1

3-bit quantization of Qwen3-1.7B achieves an average accuracy of 55.20% across five tasks, which is 18.71% higher than existing PTQ methods. These results highlight that our approach is especially effective when model capacity is constrained, such as in cases of ultra-low bit widths or limited parameter counts.

4.4 COMPARISON WITH QAT APPROACHES

This section compares our approach with two SOTA quantization-aware training (QAT) approaches: BitDistiller (Du et al., 2024) and EfficientQAT (Chen et al., 2024). To ensure a fair comparison in reasoning benchmarks, we reproduce these QAT approaches in our proposed framework. Specifically, for both methods, we perform calibration using mixed dataset, and fine-tuning with the same number of training tokens sampled from OpenThoughts-1.2M Datasets. As shown in Table 3, our approach achieves better performance in both 2-bit and 3-bit than two QAT baselines. These results demonstrate that our approach offers a more effective strategy for low-bit quantization compared to conventional QAT methods.

4.5 ABLATION STUDY

There are very few quantization-aware (QAT) training approaches that can be directly compared to ours, as most existing methods target different evaluation settings. Instead, this section presents ablation studies that compare our approach with key components derived from existing QAT methods.

Effectiveness of Block-wise Calibration: The main differences between our approach and existing methods are the use of block-wise calibration before fine-tuning and the loss function. To analyze the effects of these two components, we either replace the loss function with conventional cross-entropy loss, which is basically used in QAT (Liu et al., 2025c). In these experiments, we fine-tune the 2-bit quantized Qwen-3 1.7B model for one epoch using 32K sequences with a learn-

Table 3: Comparison of SOTA quantization approaches with various bit-width on Qwen3-1.7B models. Values are % (higher is better). We define the group size as 128.

Settings		Benchmarks (%)					Avg.	
Model	Method	Bit Width (W/A)	MATH500	LiveCodeBench	MMLU-Redux	GPQA-Diamond	IFEval	Avg.
Qwen3-1.7B	FP (Baseline)	bfloat16	89.0	53.6	74.7	38.4	70.4	65.2
	EfficientQAT	3/16	80.8	29.8	66.2	30.3	60.4	53.5
	BitDistiller	3/16	59.4	10.2	56.6	17.7	52.3	39.2
	Proposal	3/16	82.7	33.0	67.7	31.7	61.0	55.2
	EfficientQAT	2/16	24.8	0.6	29.1	12.1	25.1	18.3
	BitDistiller	2/16	12.2	1.0	22.4	21.7	19.0	15.2
	Proposal	2/16	48.6	6.5	40.1	14.5	32.2	28.4

Table 4: Ablation on block-wise calibration and loss choice: “S” denotes conventional supervised fine-tuning; “R” denotes our proposed loss; and “C” indicates the use of block-wise calibration.

Training	S	R	C	C+S	C+R
MATH-500	1.4	1.60	28.57	22.70	38.13
Live Code Bench	0.0	0.0	0.47	0.00	5.75
MMLU Redux	3.54	3.53	28.57	35.78	36.64
GPQA Diamond	6.06	6.06	7.58	14.31	14.14
IFEval	10.72	12.20	24.58	23.66	31.61

Table 5: Benchmark comparison of our proposal with INT2 Qwen3 family [quantized with our proposal](#) and BitNet b1.58 2B. We represent Instruct Strict as IS.

Benchmark (Metric)	Qwen3 8B-int2 (Ours)	Qwen3 4B-int2 (Ours)	Qwen3 1.7B-int2 (Ours)	BitNet b1.58 2B
Training Tokens	328M	328M	968M	4T
Activation	bf16	bf16	bf16	int8
MATH-500	80.13	77.13	48.60	43.40
GSM8K	88.93	81.71	57.47	58.38
IFEval (IS)	59.33	50.09	45.29	53.48
Average	76.13	69.91	50.75	51.75

ing rate of $5e-6$. Table 4 summarizes the results. Here, "S" denotes the conventional supervised fine-tuning with cross entropy loss function, "R" denotes the teacher-guided reward rectification loss in Section 3.2, and "C" means the existence of a calibration stage. Therefore, "C+S" denotes supervised fine-tuning after calibration. As shown in Table 4, both calibration data and proposed loss function significantly enhance model performance. We also find that modifying the supervised loss led to substantial improvements on reasoning benchmarks. In particular, on MATH-500, the accuracy increased by 15.43% when moving from cross entropy loss to our proposed loss, and on Live Code Bench, it increased by 5.75%. More importantly, the performance on reasoning benchmarks degrades after conducting supervised fine-tuning with cross-entropy loss. These results indicate that conventional QAT approaches, which rely primarily on cross-entropy loss for supervised fine-tuning, are insufficient for post-trained LLMs.

Effectiveness of Loss Weighting: This section studies the contribution of two loss terms, the teacher-guided reward-rectification loss and the KL-divergence loss, to overall performance. We fine-tune a 3-bit quantized Qwen3-1.7B for a single epoch with different α and β . We evaluate three different weighting schemes: $(\alpha, \beta) = (1, 0)$, which applies only the reward rectification loss; $(\alpha, \beta) = (0, 1)$, which applies only the KL divergence loss; and $(\alpha, \beta) = (0.2, 1)$ (i.e., our proposed configuration), which combines both losses with the specified weights. Table 6 demonstrates that combining the two losses improves overall model performance. **These results demonstrate that our approach exhibits stronger gains in more aggressive quantization settings, particularly at 2-bit.**

4.6 COMPARISON WITH BITNET1.58 2B4T

This section compares our quantized models with BitNet1.58 2B4T, a native ternary LLM trained from scratch. To align the bit-width, our QAT pipeline quantizes Qwen3 models into 2 bits.

Table 5 describes the accuracies on two mathematical benchmarks including MATH-500 and GSM8K, and IFEval. We referred to the results of BitNet1.58 2B4T from (Ma et al., 2025). As shown in Table 5, our INT2 quantized model achieves superior mathematical performance with lower parameter requirements and significantly fewer tokens required for the quantization process. These results demonstrate that by designing an appropriate QAT pipeline, it is possible to leverage

432 Table 6: **Effect of the loss function on reasoning performance. Comparison of 3-bit and 2-bit quan-**
 433 **titization on Qwen3 1.7B.**

435 Bit Width	(α, β)	MATH-500	Live Code Bench	MMLU Redux	GPQA Diamond	IFEval
436 3	(1.0, 0.0)	78.2	28.5	66.1	21.7	60.6
	(0.0, 1.0)	82.8	32.4	67.0	30.3	63.0
	(0.2, 1.0)	82.7	33.0	67.7	31.7	62.1
439 2	(1.0, 0.0)	22.8	1.5	32.7	15.7	25.0
	(0.0, 1.0)	2.2	0.0	6.6	2.0	8.0
	(0.2, 1.0)	48.6	6.5	40.1	14.5	32.2

444 pre-trained features, leading to promising reasoning performance train high-accuracy 2-bit models
 445 at a fraction of the training costs.

446 In addition, our approach demonstrates superior scalability compared to BitNet 1.58 2B4T. Because
 447 we fine-tune pre-trained LLMs using only a limited number of sequences, we can easily produce
 448 models with different parameter counts. This enables a flexible trade-off between performance and
 449 resource usage, as illustrated in Table 5, which demonstrates results of several parameter variations
 450 of our quantized models.

453 5 RELATED WORKS

455 In this section, we briefly summarize the quantization approaches. Quantization approaches can be
 456 categorized into post-training quantization (PTQ) and quantization-aware training (QAT) depending
 457 on whether fine-tuning is performed or not. This section deals with weight-only quantization of
 458 large language models (LLMs) addressed in this work.

459 **Post-training quantization** (PTQ) converts full-precision weights into lower-bit counterparts with-
 460 out relying on fine-tuning. To obtain better quantization parameters, recent methods optimize the
 461 reconstruction problem either at the linear projection level (Frantar et al., 2022; Lin et al., 2024)
 462 or at the transformer block level (Lee et al., 2023; Shao et al., 2023). While PTQ has achieved
 463 strong initial success in LLMs, initial approaches still face limitations in achieving extremely low-bit
 464 quantization without losing their performance. To overcome these challenges, research has shifted
 465 toward more aggressive quantization, such as 3-bit or 2-bit. Some approaches target such low-bit
 466 quantization with integer representation (Shao et al., 2023; Zhao et al., 2024; Chee et al., 2023),
 467 demonstrating noticeable performance at these bit-widths. To further improve the trade-offs be-
 468 tween accuracy and model size, recent approaches introduce vector quantization (Egiazarian et al.,
 469 2024; Tseng et al., 2024; Malinovskii et al., 2024). Despite their promising performance, vector
 470 quantization introduces substantial overhead in inference (Gong et al., 2024).

471 **Quantization-aware training** (QAT), in contrast, can enhance quantized model performance by
 472 incorporating fine-tuning. With the additional computational cost for fine-tuning, QAT enables the
 473 use of hardware-friendly numerical representations, such as integers, for low-bit quantization, re-
 474 sulting in minimal overhead at inference time. There are several choices for optimization targets
 475 for fine-tuning. LLM-QAT (Liu et al., 2023) and BitDistiller (Du et al., 2024) explore knowledge
 476 distillation within QAT literature. BitNet b1.58 (Ma et al., 2024), Spectra (Kaushal et al., 2024), and
 477 ParetoQ (Liu et al., 2025c) employ fine-tuning in a self-supervised manner using pre-training data.
 478 By spending billions of tokens for fine-tuning, these approaches realize promising performance with
 479 ternary or 2-bit. Given the substantial training costs of these approaches, recent work has focused
 480 on improving the training efficiency of QAT approaches. EfficientQAT (Chen et al., 2024) intro-
 481 duces two two-stage pipeline that perform end-to-end backpropagation following block-wise cali-
 482 bration. UPQ (Lee et al., 2025) modifies the two-stage QAT pipeline to use knowledge distillation
 483 and progressive quantization, demonstrating the promising performance on instruction-tuned LLMs.
 484 However, most existing quantization approaches have primarily focused on pre-training LLMs, with
 485 limited exploration of their effectiveness on complex reasoning capabilities that are crucial for mod-
 486 ern LLM applications. In this paper, we investigate how quantization affects reasoning performance
 487 and propose methods to preserve reasoning capabilities in quantized LLMs.

486 **Quantization and Reasoning.** Several comprehensive analyses have explored the effects of quantization
 487 on reasoning capability. Li et al. (2025) and Liu et al. (2025b) demonstrate that ultra-low-bit
 488 quantization leads to severe performance drops on reasoning benchmarks such as mathematical
 489 tasks. Liu et al. (2025b) demonstrates that less-than-4-bit quantization leads to severe performance
 490 drops on reasoning benchmarks such as mathematical tasks. Mekala et al. (2025) systematically
 491 analyzes the effects of quantization on long-context reasoning tasks, demonstrating that even 4-
 492 bit models incur substantial losses. Although these analyses reveal critical challenges for existing
 493 quantization approaches, few studies have explored effective strategies to maintain reasoning per-
 494 formance under such aggressive settings. However, one notable example is BitNet 2B4T (Ma et al.,
 495 2025), which demonstrates strong performance on mathematical benchmarks with ternary LLMs
 496 by performing quantization-aware training over four trillion tokens. In this paper, we explore a
 497 more efficient approach for reasoning-oriented LLMs. By combining block-wise quantization with
 498 RL-inspired fine-tuning using limited tokens, we obtain highly accurate 2- and 3-bit LLMs with
 499 significantly fewer fine-tuning sequences.

500 6 CONCLUSION

501 This paper addresses the critical challenge of maintaining reasoning capabilities in ultra-low-bit
 502 quantized large language models (LLMs). Through systematic analysis, we demonstrate that quan-
 503 tization affects different knowledge domains unevenly—while pre-training knowledge remains ro-
 504 bust, reasoning capabilities show severe degradation. Building on this insight, we develop a
 505 novel two-stage quantization-aware training pipeline specifically designed for post-trained reason-
 506 ing LLMs. Our approach combines mixed-domain calibration with teacher-guided reward rectifi-
 507 cation loss to preserve and restore reasoning abilities under aggressive quantization. Experiments
 508 across five reasoning benchmarks validate our method, with 2-bit quantized Qwen3-8B achieving
 509 50.45% average improvement over existing approaches. Notably, our method outperforms BitNet
 510 2B4T on mathematical reasoning while requiring dramatically fewer training resources—40K se-
 511 quences versus 4 trillion tokens. We establish the first quantization framework specifically targeting
 512 reasoning-oriented LLMs, providing practical solutions for efficient model compression without
 513 sacrificing cognitive capabilities. This work provides a foundation for future developments in effi-
 514 cient, high-performance quantized reasoning models, enabling broader deployment of sophisticated
 515 AI systems.

516 REFERENCES

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

521 Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. Quip: 2-bit quantization
 522 of large language models with guarantees. *Advances in Neural Information Processing Systems*,
 523 36:4396–4429, 2023.

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

528 Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V
 529 Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation
 530 model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

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

534 Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. Bitdistiller:
 535 Unleashing the potential of sub-4-bit llms via self-distillation. *arXiv preprint arXiv:2402.10631*,
 536 2024.

538 Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Al-
 539 istarh. Extreme compression of large language models via additive quantization. *arXiv preprint*
arXiv:2401.06118, 2024.

540 Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. Gptq: Accurate post-training
 541 quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.
 542

543 Ruihao Gong, Yang Yong, Shiqiao Gu, Yushi Huang, Yunchen Zhang, Xianglong Liu, and Dacheng
 544 Tao. Llm-qbench: A benchmark towards the best practice for post-training quantization of large
 545 language models. *CoRR*, 2024.

546 Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna
 547 Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu
 548 Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su,
 549 Wanja Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan
 550 Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak,
 551 Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia
 552 Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill,
 553 Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy,
 554 Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models,
 555 2025. URL <https://arxiv.org/abs/2506.04178>.

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

559

560 Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang,
 561 Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*,
 562 2024.

563 Ayush Kaushal, Tejas Vaidhya, Arnab Kumar Mondal, Tejas Pandey, Aryan Bhagat, and Irina Rish.
 564 Spectra: Surprising effectiveness of pretraining ternary language models at scale. *arXiv preprint
 565 arXiv:2407.12327*, 2024.

566

567 Jiedong Lang, Zhehao Guo, and Shuyu Huang. A comprehensive study on quantization tech-
 568 niques for large language models. In *2024 4th International Conference on Artificial Intelligence,
 569 Robotics, and Communication (ICAIRC)*, pp. 224–231. IEEE, 2024.

570 Jung Hyun Lee, Jeonghoon Kim, Se Jung Kwon, and Dongsoo Lee. Flexround: Learnable rounding
 571 based on element-wise division for post-training quantization. In *International Conference on
 572 Machine Learning*, pp. 18913–18939. PMLR, 2023.

573

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

577

578 Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang,
 579 Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann
 580 Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. [<https://huggingface.co/AI-MO/NuminaMath-CoT>] (https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

581

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

585

586 Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan
 587 Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization
 588 for on-device llm compression and acceleration. *Proceedings of machine learning and systems*,
 589 6:87–100, 2024.

590

591 Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng YU, Chun Yuan, and
 592 Lu Hou. Quantization hurts reasoning? an empirical study on quantized reasoning models.
 593 In *Second Conference on Language Modeling*, 2025a. URL <https://openreview.net/forum?id=BM192Ps5Nv>.

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

597

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

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

604

605 Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=Skq89Scxx>.

606

607 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.

608

609 Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu: the finest
 610 collection of educational content, 2024. URL <https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu>.

611

612 Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord,
 613 Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for
 614 science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521,
 2022.

615

616 Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Lifeng Dong,
 617 Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in
 618 1.58 bits. *arXiv preprint arXiv:2402.17764*, 1(4), 2024.

619

620 Shuming Ma, Hongyu Wang, Shaohan Huang, Xingxing Zhang, Ying Hu, Ting Song, Yan Xia, and
 621 Furu Wei. Bitnet b1. 58 2b4t technical report. *arXiv preprint arXiv:2504.12285*, 2025.

622

623 Vladimir Malinovskii, Denis Mazur, Ivan Ilin, Denis Kuznedelev, Konstantin Burlachenko, Kai Yi,
 624 Dan Alistarh, and Peter Richtarik. Pv-tuning: Beyond straight-through estimation for extreme
 625 llm compression. *Advances in Neural Information Processing Systems*, 37:5074–5121, 2024.

626

627 Anmol Mekala, Anirudh Atmakuru, Yixiao Song, Marzena Karpinska, and Mohit Iyyer. Does quan-
 628 tization affect models' performance on long-context tasks? *arXiv preprint arXiv:2505.20276*,
 629 2025.

630

631 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong
 632 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to fol-
 633 low instructions with human feedback. *Advances in neural information processing systems*, 35:
 634 27730–27744, 2022.

635

636 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea
 637 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances*
 638 *in neural information processing systems*, 36:53728–53741, 2023.

639

640 Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi
 641 Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. Code llama: Open foundation models for
 642 code. *arXiv preprint arXiv:2308.12950*, 2023.

643

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

646

647 Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang,
 648 Peng Gao, Yu Qiao, and Ping Luo. Omnipoint: Omnidirectionally calibrated quantization for
 649 large language models. *arXiv preprint arXiv:2308.13137*, 2023.

648 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 649 Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical
 650 reasoning in open language models, 2024. *URL* <https://arxiv.org/abs/2402.03300>, 2(3):5, 2024.
 651

652 Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#:
 653 Even better llm quantization with hadamard incoherence and lattice codebooks. *arXiv preprint*
 654 *arXiv:2402.04396*, 2024.

655 Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang,
 656 Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language
 657 models. *arXiv preprint arXiv:2310.11453*, 2023.

658 Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Y Wu, and Zhifang Sui.
 659 Math-shepherd: Verify and reinforce llms step-by-step without human annotations, 2024. *URL*
 660 <https://arxiv.org/abs/2312.08935>.
 661

662 Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du,
 663 Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint*
 664 *arXiv:2109.01652*, 2021.

665 Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu,
 666 Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the generalization of sft: A reinforcement learning
 667 perspective with reward rectification. *arXiv preprint arXiv:2508.05629*, 2025.
 668

669 Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and
 670 Wanxiang Che. Onebit: Towards extremely low-bit large language models. *Advances in Neural*
 671 *Information Processing Systems*, 37:66357–66382, 2024.

672 An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jian-
 673 hong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical
 674 expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.
 675

676 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu,
 677 Chang Gao, Chengan Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint*
 678 *arXiv:2505.09388*, 2025.

679 Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a ma-
 680 chine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

681 Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind
 682 Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and
 683 accurate llm serving. *Proceedings of Machine Learning and Systems*, 6:196–209, 2024.
 684

685 Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Lunling
 686 Wang, Zhihang Yuan, XiuHong Li, et al. A survey on efficient inference for large language
 687 models. *arXiv preprint arXiv:2404.14294*, 2024.
 688
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 700
 701

702 Table 7: Comparison of training cost and reasoning performance of Qwen3-1.7B under 3-bit and
 703 2-bit quantization. GPU hours are measured using the H100 GPU.

705 2nd Step	706 Bit Width	707 GPU Hours	708 MATH-500	709 LiveCodeBench	710 GPQA-Diamond	711 MMLU-Redux	712 IFEVAL	713 Avg.
GRPO (RL)	3	≈ 220	83	31.5	25.8	66.5	55.5	52.5
Ours	3	≈ 5	80.2	31.8	29.8	66.4	58.8	53.4
GRPO (RL)	2	≈ 190	24.4	0.5	11.6	29.5	24.0	18.0
Ours	2	≈ 5	45.0	2.7	10.6	34.9	29.2	24.5

714 Table 8: Comparison of first-step and fine-tuning (FT) cost along with downstream accuracy under
 715 3-bit and 2-bit quantization on Qwen3-1.7B. We report training time normalized to a single H100
 716 GPU.

717 Method	718 Bit Width	719 First step	720 FT	721 FT Epochs	722 FT Sequences	723 Avg. Acc.
EfficientQAT	3	≈ 4	≈ 22	1	32k	53.5
BitDistiller	3	≈ 0.25	≈ 26	1	32k	39.2
Block+GRPO	3	≈ 4	≈ 1180	1	32k	N/A
Ours	3	≈ 4	≈ 23	1	32k	55.2
EfficientQAT	2	≈ 4	≈ 66	3	32k	18.3
BitDistiller	2	≈ 0.25	≈ 78	3	32k	15.2
Block+GRPO	2	≈ 4	≈ 1016	3	32k	N/A
Ours	2	≈ 4	≈ 80	3	32k	28.4

724 A COMPARISON WITH ON-POLICY REINFORCEMENT LEARNING

725 While on-policy reinforcement learning incurs substantial training overhead due to its autoregressive
 726 text generation, it is commonly observed that such methods can generalize well on reasoning
 727 tasks by learning from the model’s own output distribution. Our approach, instead, doesn’t require
 728 autoregressive text generation because it only uses sequences sampled from datasets. Given these
 729 contrasting properties, it is important to assess whether our proposed off-policy objective can match
 730 the generalization typically associated with on-policy methods.

731 To clarify this trade-off, we conduct a direct comparison between our proposal and on-policy rein-
 732 forcement learning. Specifically, we compare GRPO (Shao et al., 2024), an SOTA on-policy RL
 733 approach, with our off-policy loss. To ensure a fair comparison, we conduct the experiment us-
 734 ing 6K sequences. Following DeepSeek-R1 (Guo et al., 2025), we adopt accuracy and formatting
 735 rewards for GRPO. We filtered the OpenThoughts-1.2M dataset to retain only samples with ex-
 736 tractable answer labels to ensure that the accuracy reward over the training dataset is well-defined
 737 before randomly sampling 6K sequences.

738 As shown in Table 7, our method achieves comparable performance at 3-bit precision and outper-
 739 forms GRPO at 2-bit precision with significantly reduced training time. These results demonstrate
 740 that our off-policy approach is both training-efficient and particularly effective in low-bit settings.

743 B TRAINING TIME ANALYSIS

744 As observed in Section 4, our approach performs well on reasoning benchmarks. Despite this
 745 promising performance, the overall training cost is also critical for practical quantization. This
 746 section compares the training time in both the first and fine-tuning stages to clarify this considera-
 747 tion.

748 Table 8 compares the required training time for Qwen3-1.7B with other QAT approaches. The results
 749 demonstrate that our method achieves a favorable trade-off between accuracy and training time, par-
 750 ticularly under aggressive low-bit quantization. Notably, in the 2-bit setting, our approach achieves
 751 over 10% higher accuracy than EfficientQAT, while maintaining comparable training efficiency.

755 C ANALYSIS OF DECODING LENGTH

756 Table 9: Generation length comparison of Qwen3 models under different quantization bit-widths.
757

758 759 760 761 762 763 764 765	766 767 768 769 770 771 772 773 774 775 776 777 778 779	780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809	MATH-500			GPQA-diamond			LiveCodeBench
			Model	Bit Width	Overall	Correct	Incorrect	Overall	Overall
Qwen3-1.7B	16	5905	4706	15610	6744	5561	7480	12329	
Qwen3-1.7B	3	7325	5059	17247	8564	7168	9201	15309	
Qwen3-1.7B	2	13803	6483	21065	14497	11539	14944	22466	
Qwen3-8B	16	5513	4923	14753	7397	6252	9235	11100	
Qwen3-8B	3	6296	5359	16776	10473	8730	11896	14824	
Qwen3-8B	2	9966	6136	25679	15738	13833	16779	20373	

Analyzing additional reasoning metrics, such as token count, is important for understanding the qualitative behavior of quantized models. Differences in token usage induced by quantization may reveal shifts in reasoning style or depth that are not captured by standard benchmark accuracies. Evaluating these aspects provides a more comprehensive view of how quantization affects the reasoning process.

This section addresses this aspect by analyzing the decoding length across several reasoning benchmarks. As shown in Table 9, we find that the number of thinking tokens increases as the number of bits decreases. This increase in token length primarily arises from more frequent self-correction behavior. In other words, quantized models tend to revisit and validate their answers more frequently, as illustrated in Table 10. These results indicate that quantization can introduce redundant corrective reasoning. Thus, mitigating such unnecessary thought processes during QAT is important for controlling the additional thinking tokens induced by quantization.

D FUTURE WORK

This section discusses several promising directions for further enhancing the quantized reasoning models.

Adaptive Mixing Ratio: Adapting the dataset mixing ratio in both the first and second stages is a promising research direction. This flexible strategy can enable more effective dataset selection tailored to each training phase, potentially improving overall model performance. In particular, dynamically adjusting the proportion of supervised, preference, or reasoning-heavy data may help stabilize training and reduce overfitting to any single data source. Developing principled methods for automatically determining these ratios could further enhance the efficiency and robustness of quantization-aware training. Additionally, adjusting the domain proportions within the reasoning dataset, such as increasing the ratio of code-related samples, may further enhance performance on more complex tasks like code generation. Exploring domain-specific mixing strategies remains an interesting avenue for future work.

Increasing Model Size: Applying our approach on over 10B models is also a promising research direction for future research. While our current experiments focus on smaller-scale models, where the performance degradation of quantization is more severe, our promising results suggest that the benefits of our method may scale favorably with increased parameters. Due to the limited availability of computational resources, it is difficult to conduct such larger-scale experiments. Nevertheless, we believe that our approach will scale effectively to larger LLMs.

Reducing Redundant Thinking: As observed in Appendix C, quantized models introduce additional reasoning tokens. From a practical perspective, this increase in tokens reduces throughput and requires larger KV-cache memory. Moreover, a quantitative analysis of generated samples reveals that quantization tends not to make models think more carefully, but instead leads them to repeatedly verify answers that were already produced. Suppressing such redundant verification behavior may enable more efficient quantized models.

810

811

812 Table 10: Reasoning text sample from 2bit Qwen3-8B quantized by our framework. The model can
813 be seen revisiting its reasoning multiple times.

814

815

<think>
 Okay, so I need to find the length of DE in this diagram. The problem
 says that $\sin D$ is 0.7. Let me try to visualize the diagram first.
 There's a right angle mark at D, E, and F, so maybe triangle D-E-F
 is a right triangle? Let me check the Asymptote code.

820

The Asymptote code defines points D, E, F. D is at $(\sqrt{51}, 7)$, E
 is at $(0, 7)$, and F is at $(0, 0)$. Then they draw D-E-F and a right
 angle mark at D, E, F. Wait, but right angles are usually marked
 at a single vertex. Maybe it's a right triangle with right angle
 at D? Let me think. If D is connected to E and F, and there's a
 right angle mark at D, E, F, that might mean that the triangle is
 right-angled at D? But then E and F would be connected. Let me
 sketch mentally.

827

Point D is at $(\sqrt{51}, 7)$, E is at $(0, 7)$, so E is directly to the
 left of D on the horizontal line $y=7$. Then F is at $(0, 0)$, which
 is directly below E on the vertical line $x=0$. So connecting D to E
 to F to D forms a triangle. Let me check coordinates:

831

- D: $(\sqrt{51}, 7)$
- E: $(0, 7)$
- F: $(0, 0)$

835

So DE is from D to E. Let me compute DE first. The coordinates of D
 are $(\sqrt{51}, 7)$ and E is $(0, 7)$. So the distance between D and E
 is $\sqrt{[(\sqrt{51}) - 0]^2 + (7 - 7)^2} = \sqrt{(\sqrt{51})^2 + 0} =$
 $\sqrt{51}$. Wait, $\sqrt{51}$ is approximately 7.07. But the problem
 says $\sin D$ is 0.7. Maybe I need to find DE using that? Wait, but
 maybe the diagram is not to scale? Or maybe there's some triangle
 where angle D is involved?

841

Wait, the Asymptote code draws D-E-F as a triangle and marks a right
 angle at D, E, and F? But (... 192 words omitted)

844

Wait, in triangle DEF, angle at D is between DE and DF. Let me
 confirm. If the right angle is (... 17 words omitted)

846

First, let me compute DE, EF, DF.

848

DE is from D to E: $\sqrt{(\sqrt{51} - 0)^2 + (7 - 7)^2} = \sqrt{51}$
 7.07.

851

EF is from E to F: $\sqrt{(0 - 0)^2 + (7 - 0)^2} = 7$.

852

DF is from D to F: $\sqrt{(\sqrt{51} - 0)^2 + (7 - 0)^2} = \sqrt{51 + 49} =$
 (... 3 words omitted)

855

So triangle DEF has sides $DE = \sqrt{51}$, $EF = 7$, $DF = 10$. Wait, but
 if the right angle is at E, then by Pythagoras, $DE^2 + EF^2$ should
 equal DF^2 . Let's check:

858

$DE^2 + EF^2 = 51 + 49 = 100$, which is equal to $DF^2 = 10^2 = 100$.
 Perfect! So (... 33 words omitted)

861

Wait, the problem says $\sin D = 0.7$. So $\sin(\text{angle } D) = 0.7$. Let me
 verify if this is consistent (... 3 words omitted)

863

864 In triangle DEF, angle D is between DE and DF. So in that triangle,
 865 $\sin(\text{angle D}) = \text{opposite side} / \text{hypotenuse}$. Wait, opposite side to
 866 angle D would be EF. Wait, in triangle DEF, angle at D: the side
 867 opposite angle D is EF, which is 7. The hypotenuse is DF = 10. So
 868 $\sin(\text{angle D}) = \text{EF} / \text{DF} = 7 / 10 = 0.7$. Oh! That's exactly 0.7. So
 869 that matches the problem statement. Therefore, DE is $\sqrt{51}$, but
 870 the problem says $\sin D = 0.7$, but DE is already given? Wait, but
 871 the problem is asking for DE? Wait, hold on, maybe I misread the
 872 problem?
 873 Wait, the problem says "Suppose $\sin D = 0.7$ in the diagram below.
 874 What is DE?" But in the Asymptote code, DE is $\sqrt{51}$. But maybe
 875 the diagram is not to scale? Wait, but the coordinates are given.
 876 Let me check again. Wait, maybe the diagram is different? Wait,
 877 the Asymptote code defines D as $(\sqrt{51}, 7)$, E as $(0, 7)$, F as
 878 $(0, 0)$. So DE is $\sqrt{(\sqrt{51} - 0)^2 + (7 - 7)^2} = \sqrt{51}$.
 879 So DE is $\sqrt{51}$. But then why is $\sin D$ given as 0.7? Because in
 880 the triangle DEF, $\sin D$ is $7/10 = 0.7$, so that's consistent.
 881 Therefore, is the answer DE = $\sqrt{51}$? But maybe the problem is
 882 not referring to triangle DEF? Maybe there is another triangle in
 883 the diagram? Wait, the Asymptote code draws D-E-F as a triangle,
 884 but there's also a right angle mark at D, E, F? Wait, maybe the
 885 diagram is different?
 886 But regardless, if the triangle DEF is right-angled at E, with DE =
 887 $\sqrt{51}$, EF = 7, DF = 10, then $\sin D$ is $7/10 = 0.7$, which matches
 888 the problem statement. Therefore, DE is $\sqrt{51}$. But maybe the
 889 problem is not referring to triangle DEF? Wait, the problem says
 890 "in the diagram below", but the Asymptote code is given. Wait,
 891 maybe there's another triangle in the diagram? Let me check again.
 892 But regardless, if the triangle DEF is right-angled at E, with DE =
 893 $\sqrt{51}$, EF = 7, DF = 10, (... 52 words omitted)
 894 Wait, the Asymptote code draws D-E-F, which is triangle DEF, and then
 895 draws a right angle mark at D, E, (... 120 words omitted)
 896 But regardless, the problem says $\sin D = 0.7$, and in triangle DEF,
 897 $\sin D$ is 0.7. Therefore, DE is (... 219 words omitted)
 898 Alternatively, maybe the diagram is not triangle DEF, but another
 899 triangle involving D and E? For example, maybe triangle
 900 D-E-something (... 83 words omitted)
 901 Wait, but the problem says "Suppose $\sin D = 0.7$ in the diagram below.
 902 What is DE?" If the diagram (... 41 words omitted)
 903
 904 pair D,E,F;
 905 F = (0,0);
 906 D = ($\sqrt{51}$,7);
 907 E = (0,7);
 908 draw(D--E--F--D);
 909 draw(rightanglemark(D,E,F,15));
 910 label("\$D\$",D,NE);
 911 label("\$E\$",E,NW);
 912 label("\$F\$",F,SW);
 913 label("\$7\$",(E+F)/2,W);
 914 So, in the diagram, there is a right angle at D, E, and F? Wait, no.
 915 The rightanglemark is drawn (... 159 words omitted)
 916
 917

```

918
919 Wait, but maybe the problem is not referring to triangle DEF, but
920 another triangle? Let me think. The Asymptote code (... 101 words
921 omitted)
922
923 But maybe the problem is not about triangle DEF? Wait, if angle D is
924 given as  $\sin D = 0.7$ , (... 96 words omitted)
925
926 Wait, let me check the problem again: "Suppose  $\sin D = 0.7$  in the
927 diagram below. What is DE?" So (... 86 words omitted)
928
929 Wait, let me think again. If angle D is in triangle DEF, with  $\sin D =$ 
930 0.7, and we can (... 171 words omitted)
931
932 Wait, maybe the diagram is not triangle DEF? Maybe there's a
933 different triangle where angle D is involved. Wait, let (... 116
934 words omitted)
935
936 Wait, maybe the problem is not referring to triangle DEF? Maybe there's
937 another triangle where DE is the hypotenuse or (... 38 words
938 omitted)
939
940 Wait, but the Asymptote code draws only D-E-F. So unless there is
941 another point, maybe the diagram is different? Wait, (... 137
942 words omitted)
943
944 But in that case, we need to know EF, DF, and angle D. Wait, but if
945 we have  $\sin D$  (... 69 words omitted)
946
947 Wait, but if we don't have other sides, how can we find DE? Wait,
948 unless there is a way to (... 181 words omitted)
949
950 But the problem says "in the diagram below", which is given by the
951 Asymptote code. Therefore, the answer is  $\sqrt{51}$ . (... 153 words
952 omitted)
953
954 Alternatively, maybe there is a misunderstanding in the problem.
955 Maybe the diagram is not triangle DEF, but another figure? For
956 (... 139 words omitted)
957
958 But maybe the problem is expecting a different approach? Let me think
959 again. If  $\sin D = 0.7$ , and we need to find DE. If we don't have
960 coordinates, how would we do it? Let me consider triangle D-E-F,
961 with right angle at E. Then, angle D is between DE and DF. So  $\sin$ 
962  $D = \text{opposite} / \text{hypotenuse} = EF / DF = 7 / 10 = 0.7$ . Therefore, if
963 we know  $EF = 7$ ,  $DF = 10$ , then DE can be found via Pythagoras:  $DE^2$ 
964  $+ 7^2 = 10^2 \Rightarrow DE^2 = 100 - 49 = 51 \Rightarrow DE = \sqrt{51}$ . Therefore,
965 regardless of  $\sin D$ , DE is  $\sqrt{51}$ . Therefore, the answer is
966  $\sqrt{51}$ . So maybe the problem is just giving  $\sin D$  to confirm
967 that the triangle is consistent, but the actual answer is
968  $\sqrt{51}$ . Therefore, the answer is  $\boxed{\sqrt{51}}$ .
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3
```