RATIO: REASONING ANALYSIS AND TOKEN-LEVEL INFERENCE OPTIMIZATION FOR QUANTIZED REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training quantization (PTQ) has become a widely adopted technique for reducing the memory footprint and inference cost of large language models (LLMs). However, recent studies reveal that when applied to reasoning models, PTQ not only degrades reasoning performance but also exacerbates overthinking, leading to longer reasoning trajectories. These issues may offset the efficiency gains expected from lower-precision inference. Existing approaches to mitigating these problems mainly rely on complex optimization procedures. More recent lightweight inference strategies instead use predefined overthinking markers, limiting their adaptability across quantized models. To address these issues, we propose Reasoning Analysis and Token-level Inference Optimization (RATIO), a framework that identifies model-specific overthinking tokens and assigns each a tailored penalty. RATIO first introduces Quantization-aware Reasoning Behavior Analysis (QRBA) to identify overthinking tokens by analyzing discrepancies between full-precision and quantized models. It then adopts Token-Specific Penalty Determination (TSPD), which leverages full-precision guidance to derive token-specific penalties without additional training. Extensive experiments show that RATIO achieves a better accuracy-efficiency trade-off than existing token-level interventions. Specifically, RATIO achieves up to 9.8 points accuracy improvement and reduces chain-of-thought (CoT) length by up to 51.3% compared with quantized baselines. We will release all the code and implementation of RATIO.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable performance across a wide range of tasks, with increasing model size playing an important role in their progress. Therefore, modern LLMs (Kimi Team, 2026; DeepSeek-AI et al., 2026; GLM-5 Team, 2026) continue to grow to further improve performance. However, this growth increases memory requirements and inference costs, which hinder the practical deployment of LLMs. Post-training quantization (PTQ) (Frantar et al., 2023; Lin et al., 2024; Ashkboos et al., 2024b; Yan et al., 2026a) mitigates these costs by representing model parameters at lower precision, reducing memory usage and per-step decoding costs. These benefits have made PTQ a widely adopted approach to LLM deployment.

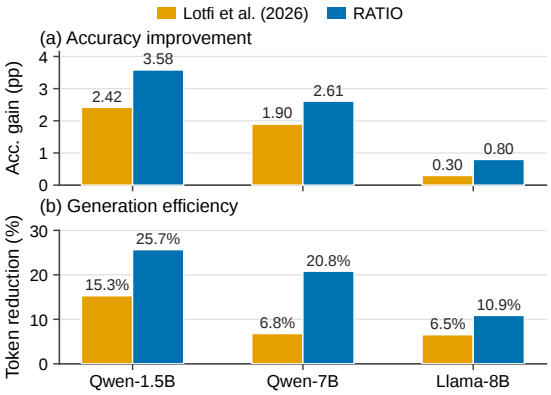


Figure 1: Performance comparison of different strategies across three quantized reasoning models. RATIO consistently achieves higher accuracy gains and larger CoT length reductions.

However, applying PTQ to reasoning models introduces new challenges, as large reasoning models (LRMs) rely on extended inference-time computation and generate long reasoning trajectories to solve complex tasks. First, low-bit quantization can introduce representation errors that degrade the final accuracy of reasoning models (Liu et al., 2025). More importantly, recent studies (Lotfi et al.,

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.



(a) Selected reasoning tokens

(b) Token-specific penalties

Figure 2: Target tokens and calibrated penalties for Qwen-1.5B. (a) Word cloud of the 21 selected tokens, repeated for visualization. (b) Token-specific logit penalties in descending order.

2026; Lian et al., 2026) have found that quantization can exacerbate overthinking behaviors, such as hesitation and repeated verification. This can lead to longer chains of thought and greater token usage, potentially offsetting the inference efficiency gains expected from quantization.

To address these issues, recent studies have explored various approaches, which can be broadly categorized into optimization-based methods and inference-time interventions. Optimization-based approaches typically adopt sophisticated strategies, such as task-specific fine-tuning (Li et al., 2025b), full-precision model guidance (Alimaskina et al., 2026), or dynamic precision adjustment (Chen et al., 2026), to recover the reasoning capability of quantized models. Recently, a lightweight decoding-time strategy (Lotfi et al., 2026) has been proposed to mitigate inefficient reasoning behaviors by penalizing manually selected overthinking markers. However, such marker-based strategies rely on predefined token lists and cannot adapt to model-specific quantization-induced reasoning changes, as overthinking tokens may vary across models. Moreover, tokens exhibit different degrees of reasoning deviation, making uniform penalties insufficient to capture token-level differences.

In this work, we propose Reasoning Analysis and Token-level Inference Optimization (**RATIO**), a framework that discovers model-specific overthinking tokens and performs token-specific calibration for quantized reasoning models. **RATIO** introduces Quantization-aware Reasoning Behavior Analysis (QRBA) to discover overthinking tokens through a two-stage process: Quantization-Sensitive Token Identification (QSTI), which identifies candidate tokens with token-level shifts, and Reasoning-context-aware Token Validation (RTV), which further selects tokens associated with inefficient reasoning behaviors based on actual quantized models' trajectories. Based on the selected tokens, Token-Specific Penalty Determination (TSPD) leverages the full-precision model as guidance to derive token-specific penalties according to token-level generation discrepancies. As shown in Fig. 2, QRBA identifies a set of model-specific reasoning tokens for Qwen-1.5B, while TSPD assigns a distinct penalty to each token. By adapting token selection and penalty strength to quantized models, **RATIO** suppresses unnecessary reasoning tokens while preserving or improving reasoning accuracy. As shown in Fig. 1, on DeepSeek-R1-Distill-Qwen-1.5B under GPTQ-W3, RATIO improves average accuracy by 3.58 percentage points and reduces CoT length by 25.7%, outperforming the best fixed-penalty baseline on both metrics.

Our main contributions are summarized as follows:

- We propose **RATIO**, a training-free framework for quantized reasoning models that enables token-level calibration without additional inference overhead.
- We introduce Quantization-aware Reasoning Behavior Analysis (QRBA), which discovers model-specific overthinking tokens through token-level discrepancy analysis and reasoning-context-aware validation on actual reasoning trajectories.
- We develop Token-Specific Penalty Determination (TSPD), which leverages the full-precision model as a reference to derive token-specific penalties and calibrates quantized models' reasoning behaviors with tailored strengths.
- Extensive experiments across multiple reasoning benchmarks, model scales and quantization settings demonstrate that **RATIO** effectively reduces unnecessary reasoning while preserving or improving the reasoning accuracy, achieving a better accuracy-efficiency trade-off over the corresponding quantized baselines.

2 Related Works

2.1 Post-Training Quantization

Post-training quantization (PTQ) has become a widely adopted technique for efficient LLM deployment. By converting high-precision parameters into low-bit representations, PTQ reduces memory and computation costs without requiring additional training. Existing PTQ methods can be broadly categorized into mixed-precision, compensation-based, and transformation-based approaches. **Mixed-precision methods** (Zhao et al., 2024; Bai et al., 2025; Huang et al., 2025; Saxena et al., 2025) adaptively allocate varying bit-widths based on quantization sensitivity. For example, Quik (Ashkboos et al., 2024a) retains outlier channels at higher precision while quantizing most weights and activations to 4 bits. **Compensation-based methods** (Li et al., 2025a; Kim et al., 2025; Arai & Ichikawa, 2025) such as GPTQ (Frantar et al., 2023) reduce reconstruction error by employing Hessian-based optimization to adjust full-precision weights. **Transformation-based methods**, including SmoothQuant (Xiao et al., 2023), AWQ (Lin et al., 2024), and QuaRot (Ashkboos et al., 2024b), apply equivalent transformations or rotations to alleviate outliers and improve low-bit quantization robustness. Beyond methods developed for integer quantization, recent works (Chen et al., 2025; Bao et al., 2026; Yan et al., 2026b) have investigated microscaling floating-point formats such as MXFP4 and NVFP4. For instance, MR-GPTQ (Egiazarian et al., 2026) further improves FP4 quantization through discrete scaling search and block-wise Hadamard rotation. However, conventional quantization evaluation mainly focuses on accuracy and per-token generation speed, which is incomplete for reasoning models because the number of generated tokens is itself a major component of the total computational cost.

2.2 Effect of quantization on reasoning

Recent studies have begun to investigate the effects of quantization on reasoning models. Empirical analyses (Li et al., 2025b; Liu et al., 2025) report degradation in reasoning accuracy and show that its severity varies with the model, quantization setting, and task. Further work (Lian et al., 2026) finds that quantization can also increase the number of reasoning tokens, even when answer accuracy is preserved. To improve the reasoning capability of quantized models, existing approaches have explored different recovery and optimization strategies. Some methods (Li et al., 2025b; Zhang et al., 2026) leverage additional training signals, such as task-specific adaptation or fine-tuning-guided quantization, to recover quantized reasoning performance. Other approaches focus on inference-time adaptation, where ASTRO (Chen et al., 2026) reduces reasoning costs through adaptive precision allocation and termination strategies, while FP16 planning and loop rescue (Alimaskina et al., 2026) mitigate generation failures in extremely low-bit reasoning models. More closely related to our work, a recent decoding-time method (Lotfi et al., 2026) reduces unnecessary reasoning by penalizing a manually specified set of overthinking markers with a shared logit bias. However, such fixed marker-based strategies overlook how strongly a token is associated with inefficient reasoning in different contexts and how much its probability shifts under quantization across models. To address this, RATIO identifies model-specific reasoning tokens and assigns tailored penalties using full-precision guidance, enabling token-specific calibration without additional training.

3 Method

3.1 Overview of RATIO

In this section, we introduce our method RATIO, as illustrated in Fig. 3. To analyze and mitigate quantization-induced changes in reasoning behavior, for each base model, RATIO compares its full-precision version with two quantized variants produced by AWQ and GPTQ. RATIO first performs Quantization-aware Reasoning Behavior Analysis (QRBA), as detailed in Sec. 3.2, to discover model-specific overthinking tokens by combining token-level discrepancies and reasoning-context information. Based on the identified tokens, RATIO further performs Token-Specific Penalty Determination (TSPD), as described in Sec. 3.3, to derive token-specific penalties for inference-time calibration under full-precision guidance.

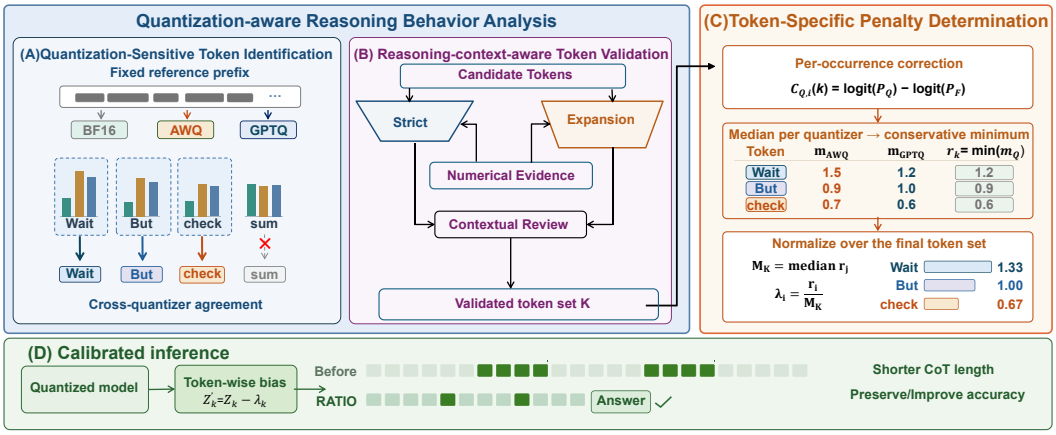


Figure 3: Overview of RATIO. The two panels on the left illustrate QRBA, which discovers model-specific quantization-sensitive reasoning tokens by combining token-level discrepancies with reasoning-context information. The rightmost panel presents TSPD, which leverages full-precision guidance to derive token-specific penalties for inference-time calibration.

3.2 Quantization-aware Reasoning Behavior Analysis

3.2.1 Quantization-Sensitive Token Identification.

To identify quantization-sensitive tokens, we perform a controlled comparison between full-precision and quantized models under identical reasoning contexts. Given a fixed reference reasoning trajectory, the full-precision model and its quantized counterparts are evaluated with the same input prefix using teacher forcing, eliminating the influence of different generation trajectories. For each reasoning position, we measure the next-token probability shift between a quantized model Q and the full-precision model F:

$$\delta_{Q,i}(k) = \log p_Q(k|x_{<t_i}) - \log p_F(k|x_{<t_i}), \tag{1}$$

where k denotes a token retained from the union of the two models' top-p candidate sets at position i, and $x_{<t_i}$ represents their shared prefix before the i-th position. Further details of the candidate sets are provided in the Appendix C.1.

For each token k, we aggregate its occurrence-level probability shifts across reasoning positions and compute the average shift separately for each quantized model:

$$\bar{\delta}_Q(k) = \frac{1}{N_Q(k)} \sum_i \delta_{Q,i}(k), \tag{2}$$

where $N_Q(k)$ denotes the number of response positions where token k is retained from the union of the top-p candidate sets for Q and F. To reduce quantizer-specific effects, we compare the average shifts from AWQ and GPTQ and retain tokens with positive shifts in both models, subject to additional support and stability criteria detailed in the Appendix C.1, for subsequent reasoning-context-aware validation. Note that this stage only captures quantization-related token-level distribution changes; the identified shifts alone do not necessarily indicate inefficient reasoning behaviors.

3.2.2 Reasoning-context-aware Token Validation

Token shifts identified on fixed reference trajectories reveal how quantization changes next-token distributions, but they do not determine whether the affected tokens are associated with inefficient reasoning behaviors. We therefore further validate the tokens obtained from QSTI using freely generated reasoning trajectories from the quantized models.

For each quantized model, we first generate reasoning trajectories on quantized models, while an independent full-precision rollout on the same questions only provides correctness information. We then compare the quantized and full-precision models on the generated quantized trajectories: the quantized model follows its own generated trajectory, while the full-precision model evaluates the same prefixes through teacher forcing. This enables token-level probability comparison under identical reasoning states while avoiding the influence of different free-generation trajectories.

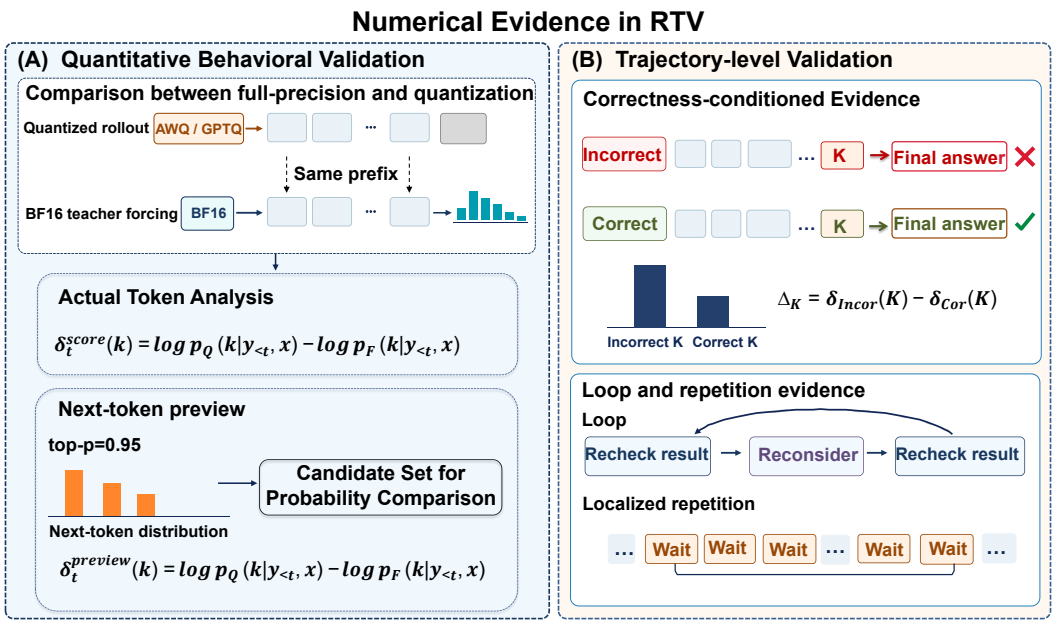


Figure 4: Numerical evidence construction in RTV. (A) Quantitative behavioral validation compares token probabilities under the same quantized prefixes. (B) Trajectory-level validation examines associations with incorrect answers and repetitive reasoning. Here, Q and F denote the quantized and full-precision models, respectively.

Quantitative Behavioral Validation. We first examine whether the quantization-related shifts identified on fixed trajectories persist during actual generation. For each token obtained from QSTI, we measure token-level probability deviations during quantized responses from two perspectives. Actual-token analysis evaluates the probability difference assigned to the token generated by the quantized model, while next-token preview analysis compares the top-p candidate distributions of the quantized and full-precision models at the same generation position. The former captures deviations of sampled tokens, while the latter captures tokens that are favored by quantization even when they are not selected during generation.

Trajectory-level Validation. Beyond quantitative behavioral validation, we further examine whether the identified tokens are associated with inefficient reasoning trajectories. Specifically, we analyze whether these tokens exhibit stronger associations with incorrect reasoning cases by comparing their occurrences between quantized-failure trajectories and successful reasoning trajectories. We additionally measure their association with repetitive reasoning patterns, including explicit loops and other forms of repetition. Together with quantitative behavioral validation, these trajectory-level statistics provide numerical evidence linking candidate tokens to inefficient reasoning outcomes. Figure 4 illustrates the construction of these numerical signals. Detailed formulations of these trajectory-level measurements are provided in the Appendix C.2.

Contextual Validation. Both preceding components rely on numerical measurements of token shifts and trajectory outcomes. However, such evidence may still be insufficient to distinguish inefficient deliberation from useful reasoning operations, such as verification and self-correction. We therefore inspect the surrounding reasoning contexts of candidate tokens to better understand their roles within reasoning trajectories. Importantly, contextual validation is only applied after tokens satisfy the corresponding numerical criteria in each selection stage. It determines token retention or exclusion but does not introduce additional tokens beyond the numerically qualified candidates.

Selection Strategy. Based on the above validation signals, Reasoning-context-aware Token Validation (RTV) adopts a staged token selection strategy. Tokens that satisfy strong quantitative behavioral criteria and sufficient support across AWQ and GPTQ are first considered as high-confidence candidates, followed by contextual validation to determine their retention. For borderline tokens with weaker quantitative behavioral evidence, additional trajectory-level and contextual validation signals are considered to determine whether they should be retained. Through this process, RTV obtains a model-specific token set for subsequent Token-Specific Penalty Determination.

3.3 Token-Specific Penalty Determination

After identifying the final quantization-sensitive reasoning token set, RATIO performs Token-Specific Penalty Determination (TSPD) to adjust the generation preference of each token during inference. Unlike marker-based approaches that apply a shared penalty, TSPD derives token-specific calibration penalties according to the deviation between quantized and full-precision models.

TSPD derives its calibration signal from the fixed reference trajectories used in QSTI. Let K denote the final model-specific token set obtained by QRBA. Given a selected token $k \in K$, the full-precision model and each quantized model are evaluated under the same reference prefix $x_{<t_i}$ using teacher forcing. For a quantized model Q and the full-precision model F, we obtain:

$$p_Q(k|x_{<t_i}), \qquad p_F(k|x_{<t_i}), \tag{3}$$

where Q denotes either the AWQ or GPTQ variant. Since TSPD only suppresses tokens whose preference is increased after quantization, we retain events satisfying:

$$p_Q(k|x_{<t_i}) > p_F(k|x_{<t_i}). \tag{4}$$

Instead of directly using probability differences, TSPD derives the correction magnitude in the logit space:

$$c_{Q,i}(k) = \mathrm{logit}(p_Q(k|x_{<t_i})) - \mathrm{logit}(p_F(k|x_{<t_i})), \tag{5}$$

where

$$\mathrm{logit}(p) = \log \frac{p}{1-p}. \tag{6}$$

The derivation of this logit-space correction is provided in the Appendix C.3. This quantity represents the token-level correction magnitude required to align the quantized preference of token k with its full-precision preference.

We use the median of correction magnitudes across occurrences to obtain the token-level correction strength, and further combine the values from AWQ and GPTQ through conservative selection:

$$m_Q(k) = \mathrm{median}_{i:p_Q(k|x_{<t_i})>p_F(k|x_{<t_i})}\, c_{Q,i}(k), \qquad r_k = \min\{m_{\mathrm{AWQ}}(k), m_{\mathrm{GPTQ}}(k)\}. \tag{7}$$

Finally, we normalize the correction magnitudes within the selected token set:

$$M_K = \mathrm{median}_{j\in K}\, r_j, \qquad \lambda_k = \frac{r_k}{M_K}. \tag{8}$$

During inference, the corresponding token logits are adjusted by:

$$z'_k = z_k - \lambda_k, \quad k \in K. \tag{9}$$

By assigning different calibration strengths to different tokens, TSPD provides a lightweight inference-time mechanism to correct quantization-induced token preference shifts.

4 Experiments

4.1 Experimental Setup

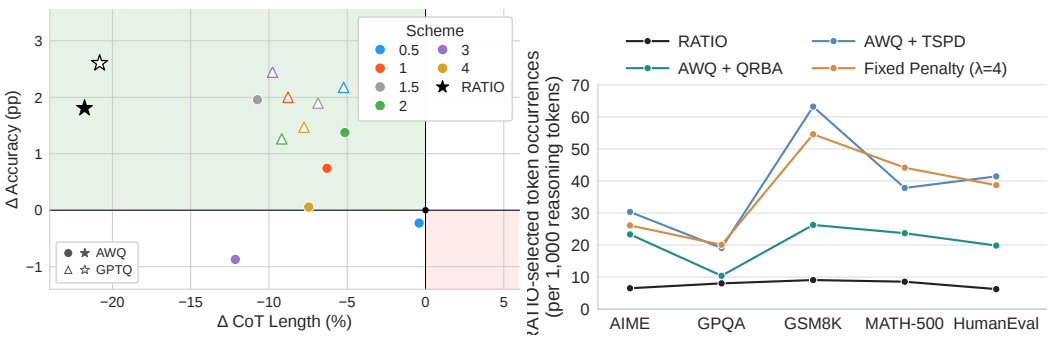
Models and Quantization Settings. We evaluate RATIO on five reasoning models, including DeepSeek-R1-Distill-Qwen-1.5B, DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Qwen-14B, DeepSeek-R1-Distill-Llama-8B, and Qwen3-4B (Yang et al., 2025) in thinking mode. The three Qwen-based distilled models and DeepSeek-R1-Distill-Llama-8B are built on the Qwen2.5 (Yang et al., 2024) and Llama 3.1 (Grattafiori et al., 2024) architectures. For each model, we consider the full-precision BF16 model and two representative post-training quantization settings: AWQ (Lin et al., 2024) and GPTQ (Frantar et al., 2023). Both AWQ and GPTQ use 3-bit weight quantization with a group size of 128. Additional details about quantization configurations are provided in the Appendix B.2.

Dataset Analyses. We collect reasoning trajectories from several reasoning datasets for token identification and validation. For QSTI, we use fixed MATH-CoT reference trajectories (Hendrycks et al., 2021) to provide controlled reasoning contexts. For RTV, we generate reasoning trajectories on AIME (Dekoninck et al., 2026), GPQA-Diamond (Rein et al., 2024), MATH-500 (Lightman et al., 2024), and GSM8K (Cobbe et al., 2021), with 50 questions randomly selected from each benchmark for each quantized model. The analysis subsets of MATH-500 and GSM8K are sampled from their corresponding training splits. During RTV, the models are provided only with the input questions, without access to any reference answers.

Table 1: Accuracy (%) and CoT length (k tokens) for BF16 and W3-quantized models using GPTQ or AWQ, with shared-penalty decoding (+ Lotfi et al. (2026)) or our method (+RATIO). The Δ columns report the accuracy change in percentage points and the relative CoT length change (%) compared with the corresponding quantized model without intervention. For each model, the fixed-penalty baseline uses the penalty strength that achieves the highest mean accuracy across AWQ and GPTQ, averaged over the five benchmarks.

		AIME	MATH	GSM8K	GPQA	HumanEval	Avg.	ΔAcc	ΔLen%
		Acc (%) / Len (k)					*Acc / Len*	*vs. baseline*	
Qwen-1.5B	**BF16**	**20.83 / 25.11**	**85.60 / 6.06**	**84.69 / 2.81**	**36.36 / 9.97**	**73.17 / 6.50**	**60.13 / 10.09**	–	–
	GPTQ	4.17 / 51.53	54.00 / 24.27	68.76 / 9.54	31.82 / 23.67	10.37 / 39.31	33.82 / 29.66	–	–
	+ Lotfi et al. (2026)	4.17 / 47.53	58.20 / 18.48	70.81 / 6.22	27.27 / 21.14	20.73 / 32.16	36.24 / 25.11	+2.42	−15.34
	+RATIO	5.00 / 42.55	61.40 / 14.06	71.49 / 3.11	30.81 / 20.64	18.29 / 29.90	37.40 / 22.05	+3.58	−25.66
	AWQ	7.50 / 55.53	44.80 / 36.13	61.18 / 22.16	23.23 / 36.56	17.68 / 39.81	30.88 / 38.04	–	–
	+ Lotfi et al. (2026)	6.67 / 41.60	56.20 / 19.36	68.54 / 8.17	28.28 / 28.76	29.27 / 31.43	37.79 / 25.86	+6.91	−32.02
	+RATIO	6.67 / 32.52	62.80 / 13.86	72.18 / 3.63	26.26 / 19.87	35.37 / 22.68	40.66 / 18.51	+9.78	−51.34
Qwen-7B	**BF16**	**42.50 / 14.27**	**93.60 / 3.83**	**91.96 / 1.50**	**52.02 / 8.60**	**85.98 / 4.87**	**73.21 / 6.61**	–	–
	GPTQ	25.00 / 33.84	86.80 / 11.06	88.63 / 3.55	39.39 / 11.50	71.95 / 9.49	62.35 / 13.89	–	–
	+ Lotfi et al. (2026)	24.17 / 34.57	84.80 / 9.25	88.25 / 2.43	42.93 / 11.74	81.10 / 6.70	64.25 / 12.94	+1.90	−6.84
	+RATIO	25.83 / 29.15	89.60 / 6.30	88.40 / 2.37	45.96 / 9.81	75.00 / 7.36	64.96 / 11.00	+2.61	−20.81
	AWQ	30.83 / 19.91	87.80 / 6.40	89.69 / 2.33	43.43 / 9.96	76.83 / 8.34	65.72 / 9.39	–	–
	+ Lotfi et al. (2026)	31.67 / 17.62	89.00 / 5.50	90.14 / 1.85	46.46 / 9.16	81.10 / 7.76	67.67 / 8.38	+1.95	−10.76
	+RATIO	32.50 / 16.41	88.60 / 5.07	89.46 / 1.82	45.96 / 8.45	81.10 / 4.98	67.52 / 7.34	+1.80	−21.83
Qwen-14B	**BF16**	**52.50 / 13.01**	**95.20 / 3.74**	**94.47 / 1.44**	**62.12 / 8.21**	**93.29 / 3.18**	**79.52 / 5.92**	–	–
	GPTQ	36.67 / 14.92	92.20 / 4.06	92.34 / 1.42	48.48 / 7.18	93.90 / 4.23	72.72 / 6.36	–	–
	+ Lotfi et al. (2026)	43.33 / 13.27	91.00 / 3.92	92.95 / 1.30	53.54 / 7.48	95.12 / 3.77	75.19 / 5.95	+2.47	−6.45
	+RATIO	35.00 / 11.55	91.80 / 3.21	92.95 / 1.10	57.58 / 6.19	92.68 / 2.74	74.00 / 4.96	+1.28	−22.01
	AWQ	39.17 / 14.29	93.20 / 4.52	93.33 / 1.77	54.55 / 8.65	92.68 / 4.50	74.59 / 6.75	–	–
	+ Lotfi et al. (2026)	38.33 / 16.09	92.00 / 4.43	93.63 / 1.68	50.51 / 8.44	91.46 / 4.36	73.19 / 7.00	−1.40	+3.70
	+RATIO	40.83 / 14.08	92.40 / 3.65	93.48 / 1.28	55.05 / 7.32	90.85 / 3.20	74.52 / 5.91	−0.07	−12.44
Llama-8B	**BF16**	**35.83 / 16.26**	**89.00 / 4.75**	**87.87 / 1.81**	**46.97 / 9.20**	**81.10 / 3.88**	**68.15 / 7.18**	–	–
	GPTQ	15.00 / 24.57	73.60 / 8.62	74.37 / 1.73	34.85 / 10.80	70.73 / 7.99	53.71 / 10.74	–	–
	+ Lotfi et al. (2026)	15.00 / 24.81	73.20 / 6.30	75.06 / 1.38	37.88 / 10.60	68.90 / 7.11	54.01 / 10.04	+0.30	−6.52
	+RATIO	15.00 / 23.58	71.40 / 6.33	73.77 / 1.52	37.37 / 9.64	75.00 / 6.80	54.51 / 9.57	+0.80	−10.89
	AWQ	18.33 / 16.40	78.00 / 5.39	83.40 / 2.43	34.34 / 8.14	72.56 / 5.99	57.33 / 7.67	–	–
	+ Lotfi et al. (2026)	15.83 / 12.93	75.40 / 5.78	82.11 / 1.93	33.33 / 7.82	73.78 / 4.94	56.09 / 6.68	−1.24	−12.91
	+RATIO	16.67 / 17.87	76.80 / 4.95	83.32 / 2.06	36.36 / 8.03	73.78 / 3.77	57.39 / 7.33	+0.06	−4.43
Qwen3-4B	**BF16**	**71.67 / 16.91**	**97.40 / 5.35**	**94.39 / 2.32**	**52.02 / 6.60**	**90.85 / 4.60**	**81.27 / 7.15**	–	–
	GPTQ	16.67 / 32.10	81.20 / 12.05	90.52 / 3.41	35.35 / 15.30	54.88 / 20.20	55.72 / 16.61	–	–
	+ Lotfi et al. (2026)	20.00 / 31.08	80.20 / 11.70	90.90 / 3.27	32.32 / 15.19	57.93 / 18.36	56.27 / 15.92	+0.55	−4.15
	+RATIO	19.17 / 28.53	80.40 / 10.72	90.52 / 2.98	35.86 / 12.80	70.12 / 13.65	59.21 / 13.74	+3.49	−17.28
	AWQ	26.67 / 23.90	85.60 / 7.94	89.69 / 3.65	38.38 / 9.62	71.95 / 12.28	62.46 / 11.48	–	–
	+ Lotfi et al. (2026)	25.00 / 22.75	87.40 / 7.30	91.43 / 3.11	40.91 / 10.34	76.22 / 10.82	64.19 / 10.87	+1.73	−5.31
	+RATIO	26.67 / 22.36	86.60 / 7.01	90.60 / 2.95	37.88 / 8.81	76.22 / 8.89	63.59 / 10.00	+1.13	−12.89

Evaluation and baseline. We evaluate RATIO on five reasoning benchmarks, including AIME (Dekoninck et al., 2026), GPQA-Diamond (Rein et al., 2024), MATH-500 (Lightman et al., 2024), GSM8K (Cobbe et al., 2021), and HumanEval (Chen et al., 2021). We report answer accuracy to measure reasoning performance. To evaluate inference efficiency, we additionally measure the generated chain-of-thought length, defined as the number of generated tokens before the final answer. We compare RATIO with quantized baselines without calibration and the decoding-time intervention method (Lotfi et al., 2026) based on manually specified overthinking markers. For fixed-penalty baselines, the same penalty strength is applied to both AWQ and GPTQ for each model, consistent with the shared calibration setting of RATIO. All experiments use default decoding with $T = 0.6$ and top-$p = 0.95$, following the setting used by prior work (Liu et al., 2025) on quantization of reasoning models. All experiments are conducted on NVIDIA RTX A6000 GPUs, each with 48GB of memory, and use a maximum generation budget of 65,536 tokens.



(a) Accuracy–efficiency trade-off (b) Selected-token occurrence rates

Figure 5: Accuracy–efficiency trade-off and selected-token occurrence analysis. (a) Changes in average accuracy and CoT length for fixed-penalty decoding with different penalty strengths and RATIO on Qwen-7B, measured relative to the corresponding uncalibrated AWQ-W3 and GPTQ-W3 baselines. RATIO occupies the upper-left region (shorter CoT and higher accuracy) under both quantizers. (b) Benchmark-wise occurrence rates of QRBA-selected tokens on Qwen-1.5B with AWQ-W3 under RATIO, its component ablations, and the fixed-penalty baseline. RATIO consistently produces lower occurrence rates across benchmarks, showing that it effectively suppresses the selected target tokens during reasoning.

4.2 Main Results

Overall Accuracy. We first evaluate the impact of RATIO on reasoning accuracy. As shown in Table 1, low-bit quantization leads to accuracy degradation, while the effectiveness of the fixed-penalty method varies across models and quantizers. In contrast, RATIO generally preserves or improves the accuracy of quantized models. For example, on Qwen-1.5B, RATIO improves the average accuracy of AWQ-W3 from 30.88% to 40.66% and that of GPTQ-W3 from 33.82% to 37.40%. Similar improvements are observed on larger models under different quantization settings. On Qwen3-4B under GPTQ-W3, RATIO improves average accuracy from 55.72% to 59.21%. These results demonstrate that RATIO effectively mitigates quantization-induced reasoning degradation across diverse reasoning models.

Reasoning Efficiency. We further evaluate the effect of RATIO on reasoning length. As shown in Table 1, RATIO consistently reduces the average CoT length of the evaluated models under both AWQ and GPTQ. On Qwen-1.5B, for example, it achieves reductions of 51.34% under AWQ-W3 and 25.66% under GPTQ-W3 relative to their uncalibrated counterparts. Reductions are also observed across the other model scales, demonstrating the general effectiveness of RATIO in controlling reasoning length. Moreover, RATIO generally produces shorter reasoning trajectories than the fixed-penalty baseline, improving the inference efficiency of quantized reasoning models. For instance, on Qwen-7B under GPTQ-W3, RATIO reduces average CoT length by 14.99% relative to the fixed-penalty baseline while also achieving higher average accuracy. These results support RATIO's effectiveness in reducing CoT length across models and quantizers.

Overall, RATIO achieves a more favorable accuracy–efficiency trade-off for quantized reasoning models. As illustrated in Fig. 5, RATIO substantially shortens reasoning trajectories while generally preserving or improving accuracy over uncalibrated quantized models, providing a better overall balance than fixed-penalty strategies. These results demonstrate the effectiveness of RATIO for accurate and efficient quantized reasoning.

4.3 Ablation Study

Effect of QRBA. We first assess the contribution of QRBA by replacing the model-specific token set identified by QRBA with the 50 manually selected overthinking markers from prior work (Lotfi et al., 2026), while retaining TSPD. As shown in Table 2, this variant consistently underperforms the complete RATIO framework. For example, under AWQ-W3, this variant reduces average accuracy by 3.27 percentage points and increases average CoT length by 49.27% compared with the complete RATIO framework. These results demonstrate the importance of QRBA in identifying model-specific tokens that can be effectively calibrated.

Table 2: Effect of QRBA and TSPD. Component-wise ablation on Qwen-1.5B under AWQ-W3 and GPTQ-W3. QRBA applies uniform penalties to the identified tokens, while TSPD applies token-specific penalties to the manually selected markers from prior work. Each entry reports accuracy (%) and CoT length (k tokens), with Δ columns showing changes relative to the corresponding quantized baseline.

		GPQA	AIME	GSM8K	MATH	HumanEval	Avg.	ΔAcc	ΔLen%
		Acc (%) / Len (k)					*Acc / Len*	*vs. baseline*	
Qwen-1.5B	AWQ	23.23 / 36.56	7.50 / 55.53	61.18 / 22.16	44.80 / 36.13	17.68 / 39.81	30.88 / 38.04	–	–
	QRBA ($\lambda = 0.5$)	29.80 / 32.37	4.17 / 48.73	69.29 / 9.17	59.40 / 20.21	34.76 / 29.86	39.48 / 28.07	+8.60	−26.21
	QRBA ($\lambda = 1.0$)	29.29 / 26.25	4.17 / 42.11	70.36 / 4.08	62.60 / 14.10	31.10 / 26.39	39.50 / 22.59	+8.62	−40.62
	TSPD	20.71 / 31.74	7.50 / 47.74	66.19 / 10.20	59.00 / 20.21	33.54 / 28.25	37.39 / 27.63	+6.51	−27.37
	RATIO	26.26 / 19.87	6.67 / 32.52	72.18 / 3.63	62.80 / 13.86	35.37 / 22.68	40.66 / 18.51	+9.78	−51.34
	GPTQ	31.82 / 23.67	4.17 / 51.53	68.76 / 9.54	54.00 / 24.27	10.37 / 39.31	33.82 / 29.66	–	–
	QRBA ($\lambda = 0.5$)	30.30 / 24.70	5.83 / 48.91	72.18 / 4.79	57.20 / 18.74	17.68 / 31.72	36.64 / 25.77	+2.82	−13.12
	QRBA ($\lambda = 1.0$)	30.30 / 22.81	6.67 / 40.04	72.25 / 3.13	61.00 / 15.29	23.17 / 28.29	38.68 / 21.91	+4.86	−26.13
	TSPD	33.84 / 19.76	2.50 / 46.07	71.04 / 5.93	57.80 / 19.63	19.51 / 32.20	36.94 / 24.72	+3.12	−16.66
	RATIO	30.81 / 20.64	5.00 / 42.55	71.49 / 3.11	61.40 / 14.06	18.29 / 29.90	37.40 / 22.05	+3.58	−25.66

Effect of TSPD. We then replace TSPD with a uniform penalty applied to all tokens identified by QRBA. Overall, the uniform-penalty variants yield a less favorable accuracy–efficiency trade-off than the complete RATIO framework, and their performance varies between AWQ and GPTQ. For example, under AWQ-W3, the strongest uniform-penalty variant reduces average accuracy by 1.16 percentage points and increases average CoT length by 22.04% compared with the complete RATIO framework. Also, a uniform penalty of $\lambda = 1$ yields lower accuracy and longer CoT than RATIO under AWQ-W3, but higher accuracy and slightly shorter CoT under GPTQ-W3. These results show that token-specific penalties provide more reliable calibration than applying a shared penalty strength to all selected tokens.

Overall, these results highlight the complementary roles of model-specific token selection and token-specific penalty determination. As shown in Fig. 5(b), RATIO produces lower occurrence rates of the selected tokens than either component ablation and the fixed-penalty baseline across all five benchmarks, further illustrating their combined effect on reasoning behavior.

5 Discussion and Future Works

While RATIO assigns token-specific penalties based on model-level quantization behaviors, these penalties remain static during inference. However, the same token may play different roles under different reasoning states, and a fixed penalty may not always be optimal for every occurrence. A promising future direction is to develop dynamic token calibration strategies that adapt the penalty strength according to the current reasoning context. For example, penalties can be relaxed during early occurrences of a token to preserve useful reasoning behaviors, while being strengthened when the token appears repeatedly within a short context window to suppress potential repetitive reasoning patterns. This would allow calibration to distinguish useful self-correction from repeated reconsideration that contributes little to solving the problem. Such state-aware calibration may further improve the balance between reasoning efficiency and model capability.

6 Conclusion

In this paper, we investigate the challenges of applying post-training quantization to reasoning models, and focus on the limitations of fixed token-level interventions. To address these issues, we propose RATIO, a framework that identifies model-specific target tokens and assigns each a tailored penalty. RATIO introduces Quantization-aware Reasoning Behavior Analysis (QRBA) to identify quantization-sensitive reasoning tokens through token-level distribution analysis and reasoning-context validation, and further develops Token-Specific Penalty Determination (TSPD) to derive token-specific penalties with full-precision guidance. Extensive experiments across multiple reasoning benchmarks and quantization settings demonstrate that RATIO effectively reduces reasoning length while preserving or improving accuracy, achieving a favorable accuracy-efficiency trade-off compared with existing token-level intervention strategies. This work establishes a behavior-aware calibration framework for quantized reasoning models, advancing efficient low-bit reasoning and providing a foundation for future research.

AI USE STATEMENT

We used generative AI tools to assist with code development and improve the manuscript's language. The authors prepared the key tables, figures, and conclusions independently. All AI-supported text and code were reviewed and verified by the authors, who take full responsibility for the final manuscript and associated artifacts.

ETHICS STATEMENT

This work studies inference-time token-level calibration for improving the efficiency of quantized reasoning models. Quantized reasoning models may retain the biases and potential for misuse of their pretrained counterparts, while reduced reasoning costs may broaden their accessibility. Our evaluations focus on reasoning accuracy and inference efficiency and do not establish the safety of the quantized reasoning models.

REPRODUCIBILITY STATEMENT

To support independent replication, we will make our code and model weights publicly available. The appendix and section 4.1 will document the experimental settings and implementation details underlying our results.

REFERENCES

Ekaterina Alimaskina, Darya Rudas, Denis Shveykin, Gleb Molodtsov, Pavel Vasiliev, and Aleksandr Beznosikov. Extreme low-bit inference in reasoning models: Failure modes and targeted recovery. *arXiv preprint arXiv:2606.02011*, 2026.

Yamato Arai and Yuma Ichikawa. Quantization Error Propagation: Revisiting Layer-Wise Post-Training Quantization. In *NeurIPS*, 2025.

Saleh Ashkboos, Ilia Markov, Elias Frantar, Tingxuan Zhong, Xincheng Wang, Jie Ren, Torsten Hoefler, and Dan Alistarh. Quik: Towards end-to-end 4-bit inference on generative large language models. In *EMNLP*, 2024a.

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*, 2024b.

Runsheng Bai, Bo Liu, and Qiang Liu. Skim: Any-bit quantization pushing the limits of post-training quantization. In *ICML*, 2025.

Chengzhu Bao, Xianglong Yan, Zhiteng Li, Guangshuo Qin, Guanghua Yu, and Yulun Zhang. Soar: Scale optimization for accurate reconstruction in nvfp4 quantization. *arXiv preprint arXiv:2605.12245*, 2026.

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

Tianle Chen, Pengyu Cheng, Qiyuan Zhu, Jiacheng Wang, Bei Liu, Hao Gu, Ruijie Shen, Xiaofeng Hou, Sirui Han, and Jiacheng Liu. Adaptive spatial and temporal redundancy optimization for efficient reasoning in large language models. In *ACL*, 2026.

Yuzong Chen, Xilai Dai, Jake Hyun, Chi-Chih Chang, Wonsuk Jang, Yuheng Wu, Thierry Tambe, Jae sun Seo, and Mohamed S. Abdelfattah. Razer: Pushing the limits of nvfp4 quantization with redundant zero remapping. *arXiv preprint arXiv:2501.04052*, 2025.

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

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 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.

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

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. In *ICLR*, 2023.

GLM-5 Team. GLM-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. SliM-LLM: Salience-Driven Mixed-Precision Quantization for Large Language Models. In *ICML*, 2025.

Junhan Kim, Ho-young Kim, Eulrang Cho, Chungman Lee, Joonyoung Kim, and Yongkweon Jeon. BOA: Attention-aware Post-training Quantization without Backpropagation. In *ICML*, 2025.

Kimi Team. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. GPTAQ: Efficient Finetuning-Free Quantization for Asymmetric Calibration. In *ICML*, 2025a.

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

Xinyu Lian, Walid Krichene, Beichen Huang, Masahiro Tanaka, Olatunji Ruwase, Li Zhang, and Minjia Zhang. Quantization inflates reasoning: Token inflation as a hidden cost of low-bit reasoning models. In *EMNLP*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. In *MLSys*, 2024.

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

Sanae Lotfi, Polina Kirichenko, Steven Li, and Zechun Liu. Quantized reasoning models think they need to think longer, but they do not. In *NeurIPS*, 2026.

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. In *COLM*, 2024.

Utkarsh Saxena, Sayeh Sharify, Kaushik Roy, and Xin Wang. Resq: Mixed-precision quantization of large language models with low-rank residuals. In *ICML*, 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. In *ICML*, 2025.

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*, 2023.

Xianglong Yan, Chengzhu Bao, Zhiteng Li, Tianao Zhang, Shaoqiu Zhang, Ruobing Xie, Samm Sun, and Yulun Zhang. D^2Quant: Accurate low-bit post-training weight quantization for llms. In *NeurIPS*, 2026a.

Xianglong Yan, Hong Liu, Chengzhu Bao, Tianao Zhang, Guanghua Yu, Jianchen Zhu, and Yulun Zhang. Focus: Fp4 optimization via coupled-relaxation and dual-granularity scaling. *arXiv preprint arXiv:2608.01847*, 2026b.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

Nan Zhang, Eugene Kwek, Yusen Zhang, Muyu Pan, Suhang Wang, Prasenjit Mitra, and Rui Zhang. Quantlrm: Quantization of large reasoning models via fine-tuning signals. *arXiv preprint arXiv:2602.02581*, 2026.

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving. In *MLSys*, 2024.

A More Experimental Results

A.1 Performance on Other Quantization Methods

To evaluate whether RATIO remains effective beyond weight-only quantization, we further apply it to DeepSeek-R1-Distill-Qwen-1.5B quantized with FlatQuant (Sun et al., 2025) under the W4A4KV4 setting. As shown in Table 3, RATIO improves average accuracy from 45.27% to 48.48% while reducing average CoT length by 12.90%. It also provides a more favorable accuracy–efficiency trade-off than fixed-penalty decoding, demonstrating its applicability to the joint low-bit quantization of weights, activations, and the KV cache.

Table 3: Performance of RATIO on Qwen-1.5B with FlatQuant W4A4KV4. Accuracy (%) and CoT length (k tokens) are reported as Acc/Len. The Δ columns report changes relative to uncalibrated FlatQuant. For each metric, the highest accuracy and shortest CoT length are in bold.

		GPQA	AIME	GSM8K	MATH	HumanEval	Avg.	ΔAcc	ΔLen%
		Acc (%) / Len (k)					*Acc / Len*	*vs. baseline*	
Qwen-1.5B	FlatQuant	**34.34** / 25.68	8.33 / **37.20**	74.37 / 4.06	65.40 / 16.47	43.90 / 14.64	45.27 / 19.61	–	–
	+ Lotfi et al. (2026)	30.30 / **20.81**	**11.67** / 39.26	**75.66** / 2.83	69.80 / 13.30	48.17 / **11.75**	47.12 / 17.59	+1.85	−10.30
	+RATIO	30.81 / 22.16	**11.67** / 37.77	75.51 / **1.69**	**73.20** / **11.87**	**51.22** / 11.90	**48.48** / **17.08**	**+3.21**	**−12.90**

B Additional Experimental Details

B.1 Selected Tokens and Penalties

Tables 4 and 5 report the final model-specific token sets produced by QRBA and their normalized TSPD penalties. The same model-specific token set and penalty values are applied to its AWQ-W3 and GPTQ-W3 variants.

Table 4: Model-specific tokens and normalized penalties used by RATIO. A leading underscore denotes a whitespace prefix in the decoded token.

Qwen-1.5B		Qwen-7B		Llama-8B		Qwen-14B	
Token	λ_k	**Token**	λ_k	**Token**	λ_k	**Token**	λ_k
_but	1.22	_but	1.06	_back	1.00	_but	1.00
_what	1.21	_back	0.58	_try	0.90	_what	1.04
_try	0.73	_check	0.48	_well	0.86	_think	0.71
_again	1.16	_But	1.15	_check	0.93	_But	1.23
_well	0.88	_something	1.09	_But	1.12	_something	1.57
_But	1.09	_might	0.75	_something	1.34	_question	0.80
_something	1.12	_question	1.32	_thought	0.85	_actually	1.05
_might	1.03	_wait	0.92	_question	1.10	_wait	0.92
No	0.74	But	1.00	_actually	0.95	But	1.14
_thought	0.78	_trying	0.64	_wait	0.96	_correct	0.92
But	1.09	_ensure	0.67	But	1.07	_trying	0.71
_correct	1.00	but	1.04	_correct	1.13	_seems	0.88
_ensure	1.00	_Wait	1.18	_seems	1.11	_maybe	0.77
Yes	0.90	Wait	1.12	_maybe	0.93	_confirm	0.69
However	1.04	_Thus	0.37	_perhaps	0.96	_perhaps	0.96
_Wait	0.96	Thus	0.36	_Wait	1.14	_Wait	1.33
Wait	1.12	Hmm	1.03	something	1.29	Wait	1.32
_conclude	1.02					something	1.78
Thus	0.48					Thus	0.36
Hmm	0.97					Hmm	1.23
Alternatively	0.97					_Hmm	1.25

Table 5: Selected tokens and normalized penalties for Qwen3-4B in thinking mode. A leading underscore denotes a whitespace prefix in the decoded token.

Token	λ_k	Token	λ_k
_but	1.11	_question	1.11
_what	1.03	But	1.06
_back	0.83	_However	0.94
_think	0.81	but	1.00
_But	1.05	_perhaps	0.84
_something	1.00	Wait	1.02
_looking	0.82	_mistake	0.79

B.2 Quantization and Implementation Details

Quantization Settings. We provide the detailed quantization and inference configurations used in our experiments. For AWQ, we follow the standard AWQ recipe and use 128 sequences with a sequence length of 512 sampled from the Pile validation set as the calibration data. For GPTQ, we use WikiText-2 as the calibration dataset, with 128 sequences of length 2048. All quantized models use a group size of 128 for weight quantization.

Implementation Details. Unlike the 65,536-token generation budget specified in the main text, Qwen3-4B is subject to a total context limit of 40,960 tokens.

C Additional Method Details

C.1 Candidate-Set Construction and QSTI Criteria

Candidate-Set Construction. At each reasoning position i, we first construct a raw candidate pool from the union of the top-p candidate sets produced by the quantized model Q and the full-precision model F:

$$\mathcal{U}_{Q,i} = \mathcal{N}_{Q,i} \cup \mathcal{N}_{F,i}, \tag{10}$$

where $\mathcal{N}_{Q,i}$ and $\mathcal{N}_{F,i}$ denote the corresponding top-p candidate sets under the shared prefix $x_{<t_i}$. We set $p = 0.95$, consistent with the decoding configuration used in our other experiments. For tokens appearing in both candidate sets, we directly use the probabilities produced by the two models. For a token appearing in only one candidate set, we use a probability floor $\epsilon = 10^{-3}$ for the missing side.

A probability floor may introduce an unreliable shift when it is larger than the observed probability on the available side. We therefore apply a filtering criterion to single-sided candidate events. Specifically, the retained candidate set is defined as

$$\mathcal{C}_{Q,i} = (\mathcal{N}_{Q,i} \cap \mathcal{N}_{F,i}) \cup \mathcal{S}_{Q,i} \cup \mathcal{S}_{F,i}, \tag{11}$$

where

$$\mathcal{S}_{Q,i} = \{k \in \mathcal{N}_{Q,i} \setminus \mathcal{N}_{F,i} : p_Q(k|x_{<t_i}) > \epsilon\}, \tag{12}$$

and

$$\mathcal{S}_{F,i} = \{k \in \mathcal{N}_{F,i} \setminus \mathcal{N}_{Q,i} : p_F(k|x_{<t_i}) > \epsilon\}. \tag{13}$$

For each retained token $k \in \mathcal{C}_{Q,i}$, we compute the token-level probability shift as

$$\delta_{Q,i}(k) = \log p_Q(k|x_{<t_i}) - \log p_F(k|x_{<t_i}). \tag{14}$$

For a retained single-sided event, the probability on the missing side is set to ϵ only when computing this shift. For example, consider a token that appears only in the quantized candidate set and satisfies

$$p_Q(k|x_{<t_i}) = 0.003 > \epsilon. \tag{15}$$

Using $p_F(k|x_{<t_i}) = \epsilon = 0.001$ for shift estimation gives

$$\delta_{Q,i}(k) = \log 0.003 - \log 0.001 > 0. \tag{16}$$

In contrast, if the observed probability is $10^{-5} \leq \epsilon$, the event is excluded before shift estimation. In this case, assigning ϵ to the missing side would make the direction of the resulting shift primarily determined by the artificial floor rather than reliable model evidence. Only events retained in $\mathcal{C}_{Q,i}$ are included in the occurrence-level aggregation and subsequent QSTI statistics.

Support and Stability Criteria. After obtaining valid occurrence-level shifts, we further apply support and stability criteria to identify reliable quantization-sensitive tokens. For each token k, we denote $N_Q(k)$ as the number of valid shift events under quantizer Q, $S_Q(k)$ as the number of reference trajectories containing these events, and $\bar{\delta}_Q(k)$ as the average occurrence-level shift.

We first retain tokens that are observed under both AWQ and GPTQ and satisfy the token-cleaning rule. The cleaning rule removes special tokens, punctuation, non-English pieces, and uninterpretable tokenizer fragments. We then require sufficient event support from both quantizers:

$$N_{\mathrm{AWQ}}(k) > 100, \qquad N_{\mathrm{GPTQ}}(k) > 100. \tag{17}$$

To ensure that the identified preference changes are not specific to a single quantization method, we further require the average shifts of AWQ and GPTQ to have the same direction:

$$\bar{\delta}_{\mathrm{AWQ}}(k) \cdot \bar{\delta}_{\mathrm{GPTQ}}(k) > 0. \tag{18}$$

Finally, we require sufficient trajectory coverage:

$$S_{\mathrm{AWQ}}(k) > 105, \qquad S_{\mathrm{GPTQ}}(k) > 105. \tag{19}$$

The tokens satisfying these requirements form the shared candidate pool K_{base}.

We then perform stability filtering on the positive-shift candidates in K_{base}. Since RATIO aims to suppress tokens whose preference is increased by quantization, negative-shift tokens are not included in the final calibration set. For each positive token, we compute three complementary statistics. First, the conservative shift magnitude is defined as:

$$R_{\Delta}(k) = \min\left(|\bar{\delta}_{\mathrm{AWQ}}(k)|, |\bar{\delta}_{\mathrm{GPTQ}}(k)|\right). \tag{20}$$

Second, we measure directional consistency across occurrence-level events:

$$R^{+}_{\mathrm{dir}}(k) = \min\left(\mathrm{Pr}_{\mathrm{AWQ}}(\delta_i > 0), \mathrm{Pr}_{\mathrm{GPTQ}}(\delta_i > 0)\right). \tag{21}$$

Here, $\mathrm{Pr}_{\mathrm{AWQ}}(\delta_i > 0)$ represents the proportion of shift events where AWQ assigns a higher probability to token k than the full-precision model. Third, we measure the proportion of events where the token appears in both the quantized and full-precision top-p candidate sets:

$$R_{\mathrm{both}}(k) = \min\left(\mathrm{Pr}_{\mathrm{AWQ}}(k \in \mathcal{N}_{Q,i} \cap \mathcal{N}_{F,i}), \mathrm{Pr}_{\mathrm{GPTQ}}(k \in \mathcal{N}_{Q,i} \cap \mathcal{N}_{F,i})\right). \tag{22}$$

These three statistics characterize the strength, consistency, and reliability of the observed quantization-related shifts. The corresponding thresholds are determined adaptively from the lower quartiles of the positive candidate distribution:

$$\tau_{\Delta}^{+} = \mathrm{Quantile}_{0.25}(R_{\Delta}), \tag{23}$$

$$\tau_{\mathrm{dir}}^{+} = \mathrm{Quantile}_{0.25}(R_{\mathrm{dir}}^{+}), \tag{24}$$

$$\tau_{\mathrm{both}}^{+} = \mathrm{Quantile}_{0.25}(R_{\mathrm{both}}). \tag{25}$$

A token is retained as a numerically stable candidate when:

$$R_{\Delta}(k) \geq \tau_{\Delta}^{+}, \tag{26}$$

$$R_{\mathrm{dir}}^{+}(k) \geq \tau_{\mathrm{dir}}^{+}, \tag{27}$$

$$R_{\mathrm{both}}(k) \geq \tau_{\mathrm{both}}^{+}. \tag{28}$$

The resulting set is denoted as K_{num}.

To improve recall, we additionally consider a small reasoning-state lexical recovery branch within K_{base}. This branch targets tokens with strong associations with reasoning-related behaviors but weaker numerical stability. Such tokens are not directly included in the calibration set; instead, they are passed to subsequent reasoning-context-aware validation for further verification. The resulting set is denoted as K_{lex}.

The final QSTI candidate pool is defined as:

$$K_{\mathrm{QSTI}} = K_{\mathrm{num}} \cup K_{\mathrm{lex}}. \tag{29}$$

All tokens in K_{QSTI} are further validated by RTV before being used for token-specific calibration.

C.2 Reasoning-context-aware Token Validation

C.2.1 Numerical Evidence Construction

Given an input x_r, each quantized model $Q \in \{\mathrm{AWQ}, \mathrm{GPTQ}\}$ freely generates a reasoning trajectory y_r. The full-precision model F then evaluates the same quantized prefixes through teacher forcing. This allows all token-level comparisons to be performed under identical reasoning states. For each candidate $k \in K_{\mathrm{QSTI}}$, we construct the following quantitative evidence.

Actual-token Scoring. Suppose that the quantized model generates token $y_{r,t} = k$ at position t in trajectory r. We measure its probability shift under the same quantized prefix as

$$\delta_{Q,r,t}^{\mathrm{score}}(k) = \log p_Q(k|y_{r,<t}, x_r) - \log p_F(k|y_{r,<t}, x_r). \tag{30}$$

A positive value indicates that the quantized model assigns a higher probability to the generated token than the full-precision model under the same reasoning state. Let

$$\mathcal{O}_Q(k) = \{(r,t) : y_{r,t} = k\} \tag{31}$$

denote the set of actual-token events for k. We compute its event-weighted mean anomaly as

$$A_Q(k) = \frac{1}{|\mathcal{O}_Q(k)|} \sum_{(r,t)\in\mathcal{O}_Q(k)} \delta_{Q,r,t}^{\mathrm{score}}(k). \tag{32}$$

Every occurrence receives equal weight in $A_Q(k)$. Consequently, trajectories in which the token occurs frequently contribute more strongly to this statistic.

Trajectory-level Aggregation. To measure whether the observed shift is consistent across different reasoning trajectories, let

$$I_Q(k) = \{r : \exists t,\ (r,t) \in \mathcal{O}_Q(k)\} \tag{33}$$

denote the set of trajectories containing at least one actual-token event for k. Its trajectory support is defined as

$$S_Q(k) = |I_Q(k)|. \tag{34}$$

Each trajectory contributes at most one unit to $S_Q(k)$, regardless of how frequently the token occurs within that trajectory. Let $n_{Q,r}(k)$ denote the number of occurrences of k in trajectory r. We first compute its within-trajectory mean anomaly:

$$\bar{a}_{Q,r}(k) = \frac{1}{n_{Q,r}(k)} \sum_{t:y_{r,t}=k} \delta^{\text{score}}_{Q,r,t}(k), \tag{35}$$

and then average equally across trajectories:

$$T_Q(k) = \frac{1}{S_Q(k)} \sum_{r \in I_Q(k)} \bar{a}_{Q,r}(k). \tag{36}$$

Unlike the event-weighted statistic $A_Q(k)$, $T_Q(k)$ assigns equal weight to each trajectory and therefore reduces the influence of individual long or repetitive responses. We define the corresponding cross-quantizer evidence as

$$E_{\text{trajectory}}(k) = [T_{\text{AWQ}}(k) > 0] \wedge [T_{\text{GPTQ}}(k) > 0]. \tag{37}$$

This condition indicates that the positive anomaly persists across reasoning trajectories under both quantizers.

Next-token Preview. Actual-token scoring considers only tokens selected during generation. To capture tokens whose probabilities are increased by quantization even when they are not sampled, we additionally compare the top-p candidate distributions under the same quantized prefix. For a token k appearing in the quantized top-p set, its preview shift is

$$\delta^{\text{preview}}_{Q,r,t}(k) = \log p_Q(k|y_{r,<t}, x_r) - \log p_F(k|y_{r,<t}, x_r). \tag{38}$$

If k is absent from the full-precision top-p set, we use the same probability floor as in QSTI:

$$p_{\text{floor}} = 0.001. \tag{39}$$

This value is used only to approximate the missing-side probability and is unrelated to the inference-time penalty. We aggregate preview events similarly to actual-token events, obtaining trajectory support $S^{\text{preview}}_Q(k)$ and mean anomaly $A^{\text{preview}}_Q(k)$. Preview evidence is activated when

$$E_{\text{preview}}(k) = \bigwedge_{Q \in \{\text{AWQ},\text{GPTQ}\}} \left([S^{\text{preview}}_Q(k) \geq 20] \wedge [A^{\text{preview}}_Q(k) > 0] \right). \tag{40}$$

Correctness-conditioned Evidence. We use independent full-precision rollouts to divide the quantized trajectories into two primary groups:

$$\mathcal{E}_Q = \{r : Q \text{ is incorrect and } F \text{ is correct}\}, \tag{41}$$

$$\mathcal{C}_Q = \{r : Q \text{ is correct and } F \text{ is correct}\}. \tag{42}$$

The first group represents reasoning failures introduced by quantization, while the second serves as a control group. For evidence type $u \in \{\text{score}, \text{preview}\}$, let $\bar{a}^u_{Q,r}(k)$ denote the corresponding trajectory-level mean anomaly. We define the correctness-conditioned difference as

$$\Delta^u_Q(k) = \text{mean}_{r\in\mathcal{E}_Q}\, \bar{a}^u_{Q,r}(k) - \text{mean}_{r\in\mathcal{C}_Q}\, \bar{a}^u_{Q,r}(k). \tag{43}$$

The means are taken over trajectories in which the corresponding event for k is observed. A positive value indicates that the token exhibits a stronger quantization-related anomaly when the quantized model fails on a problem solved correctly by the full-precision model. We define

$$E_{\text{score-correct}}(k) = \bigwedge_{Q\in\{\text{AWQ},\text{GPTQ}\}} [\Delta^{\text{score}}_Q(k) > 0], \tag{44}$$

and

$$E_{\text{preview-correct}}(k) = \bigwedge_{Q\in\{\text{AWQ},\text{GPTQ}\}} [\Delta^{\text{preview}}_Q(k) > 0]. \tag{45}$$

The two indicators are combined as

$$E_{\text{correct}}(k) = E_{\text{score-correct}}(k) \vee E_{\text{preview-correct}}(k). \tag{46}$$

Thus, $E_{\text{correct}}(k)$ indicates that either actual-token or preview anomalies are consistently stronger in quantized-model failures than in shared successes under both quantizers.

Loop and Repetition Evidence. We construct two complementary forms of repetition evidence: token enrichment in explicit-loop trajectories and localized high-repeat patterns.

Explicit-loop detection. We first label a trajectory as an explicit loop if it satisfies

$$(\text{repeat4} \geq 80\% \wedge \text{duplicate-lines} \geq 50\%) \vee (\text{max same token run} \geq 64)\,. \tag{47}$$

Here, we enumerate all contiguous 4-token windows in the trajectory. A window is treated as repeated if its exact 4-token sequence occurs at least twice within that trajectory, and all occurrences of such a sequence, including its first occurrence, are counted as repeated windows. The value of repeat4 is the number of repeated-window occurrences divided by the total number of 4-token windows. Thus, $\text{repeat4} \geq 80\%$ means that at least 80% of the trajectory's 4-token windows have an identical counterpart elsewhere in the same trajectory. To compute duplicate-lines, we split the decoded trajectory at newline characters and discard empty lines and lines shorter than a predefined length threshold. We then compare the remaining lines using exact string matching. A retained line is counted as duplicated if an identical line appears elsewhere in the same trajectory, and duplicate-lines is the proportion of such duplicated line occurrences among all retained lines. Finally, $\text{max same token run}$ denotes the longest consecutive repetition of a single token. An explicit loop that produces an incorrect answer is assigned to the error-loop set L^{error}_Q, whereas a trajectory that does not satisfy the loop criterion is assigned to the non-loop set L^{nonloop}_Q.

Token enrichment in explicit loops. After labeling the trajectories, we determine whether candidate token k occurs more frequently in error-loop trajectories than in non-loop trajectories. For evidence type $u \in \{\text{score}, \text{preview}\}$, its length-normalized occurrence rate in trajectory r is

$$f^u_{Q,r}(k) = 1000 \frac{\text{count}^u_{Q,r}(k)}{\text{length}(y_r)}. \tag{48}$$

We then compute the loop-enrichment ratio

$$R_Q^{\text{loop},u}(k) = \frac{\text{mean}_{r \in L_Q^{\text{error}}} f_{Q,r}^u(k)}{\text{mean}_{r \in L_Q^{\text{nonloop}}} f_{Q,r}^u(k)}. \tag{49}$$

The corresponding cross-quantizer indicators are

$$E_{\text{score-loop}}(k) = \bigwedge_{Q \in \{\text{AWQ},\text{GPTQ}\}} [R_Q^{\text{loop,score}}(k) > 1], \tag{50}$$

$$E_{\text{preview-loop}}(k) = \bigwedge_{Q \in \{\text{AWQ},\text{GPTQ}\}} [R_Q^{\text{loop,preview}}(k) > 1]. \tag{51}$$

If the mean occurrence rate in non-loop trajectories is zero while that in error-loop trajectories is positive, the ratio is treated as infinity. If both rates are zero, the ratio is treated as missing and does not activate the corresponding indicator.

Localized high-repeat evidence. The trajectory-level loop criterion may overlook localized repetition. We therefore inspect all 4-token windows covering each occurrence of k. An occurrence is marked as a high-repeat event if any such exact 4-token sequence occurs at least 8 times within the same trajectory. Let $H_Q(k)$ denote the number of trajectories containing at least one high-repeat event for k. We define

$$E_{\text{repeat}}(k) = [H_{\text{AWQ}}(k) > 0] \wedge [H_{\text{GPTQ}}(k) > 0]. \tag{52}$$

Because this condition is intentionally permissive, it is used only as supporting evidence.

Evidence Consolidation. Finally, we consolidate the raw indicators into four distinct evidence categories. We retain $E_{\text{trajectory}}(k)$ and $E_{\text{preview}}(k)$ as separate categories, and define

$$E_{\text{loop/repeat}}(k) = E_{\text{score-loop}}(k) \vee E_{\text{preview-loop}}(k) \vee E_{\text{repeat}}(k). \tag{53}$$

Together with $E_{\text{correct}}(k)$ defined above, the number of distinct evidence categories is

$$N_C(k) = E_{\text{trajectory}}(k) + E_{\text{preview}}(k) + E_{\text{correct}}(k) + E_{\text{loop/repeat}}(k). \tag{54}$$

Each category contributes at most one count, preventing closely related indicators from being counted repeatedly.

C.2.2 Staged Token Selection

Based on the evidence constructed above, RTV progressively selects tokens through a strict core and two expansion stages. All stages operate exclusively on K_{QSTI}, ensuring that contextual validation cannot introduce tokens that have not passed QSTI.

Strict core. We first construct a reference pool containing candidates with sufficient trajectory support and positive actual-token anomalies under both quantizers:

$$K_{\text{ref}} = \left\{ k \in K_{\text{QSTI}} : \bigwedge_{Q \in \{\text{AWQ},\text{GPTQ}\}} [S_Q(k) \geq 20 \wedge A_Q(k) > 0] \right\}. \tag{55}$$

For each quantizer, we compute the lower quartile of its anomaly values within the reference pool:

$$\tau_Q = \text{Quantile}_{0.25}\left(\{A_Q(j) : j \in K_{\text{ref}}\}\right), \qquad Q \in \{\text{AWQ}, \text{GPTQ}\}. \tag{56}$$

A candidate passes the strict numerical gate if

$$G_{\text{strict}}(k) = [k \in K_{\text{QSTI}}] \wedge \bigwedge_{Q \in \{\text{AWQ},\text{GPTQ}\}} [S_Q(k) \geq 20 \wedge A_Q(k) \geq \tau_Q]\,. \tag{57}$$

The thresholds τ_Q are derived separately for each model and quantizer rather than shared across models. This gate therefore requires substantial positive anomalies under both AWQ and GPTQ, rather than deviations that are only marginally above zero. Candidates passing the gate are further examined using reasoning-state lexical information. A token is retained in the strict core only when its observed use is directly associated with reasoning operations such as restarting, backtracking, verification, hesitation, self-questioning, self-correction, uncertainty, or metacognitive control.

First-round expansion (R1). To recover candidates with moderate support or asymmetric anomaly strength across quantizers, R1 applies a relaxed numerical gate:

$$G_{\text{R1}}(k) = [k \in K_{\text{QSTI}}] \wedge \bigwedge_{Q \in \{\text{AWQ},\text{GPTQ}\}} [S_Q(k) \geq 10] \wedge \bigvee_{Q \in \{\text{AWQ},\text{GPTQ}\}} [A_Q(k) > 0] \wedge [N_C(k) \geq 1] \wedge \neg G_{\text{strict}}(k). \tag{58}$$

All R1 candidates are first examined using the same reasoning-state lexical information as the strict core. We additionally require contextual review when the candidate is supported by only one evidence category and that category is either trajectory-level or preview evidence. The contextual-review trigger is

$$[N_C(k) = 1] \wedge [E_{\text{trajectory}}(k) \vee E_{\text{preview}}(k)] \wedge \neg E_{\text{correct}}(k) \wedge \neg E_{\text{loop/repeat}}(k). \tag{59}$$

This condition is deterministic. It is activated because trajectory-level and preview evidence primarily reflect distributional deviations; without correctness-conditioned or repetition-related support, the surrounding reasoning context must be inspected to distinguish inefficient reasoning from valid verification or self-correction.

Second-round expansion (R2). R2 considers near-miss candidates with weaker numerical support but evidence from multiple complementary categories:

$$G_{\text{R2}}(k) = [k \in K_{\text{QSTI}}] \wedge \bigwedge_{Q \in \{\text{AWQ},\text{GPTQ}\}} [S_Q(k) \geq 5] \wedge [N_C(k) \geq 2] \wedge \neg G_{\text{strict}}(k) \wedge \neg G_{\text{R1}}(k). \tag{60}$$

Because R2 uses the most relaxed support threshold, every candidate passing this gate undergoes contextual review. We inspect its occurrences in the original quantized trajectories to determine whether the token is consistently associated with inefficient deliberation, rather than necessary verification, correction, or other valid reasoning behavior.

Let K_{strict}, K_{R1}, and K_{R2} denote the tokens retained by the three stages. The final model-specific token set is

$$K = K_{\text{strict}} \cup K_{\text{R1}} \cup K_{\text{R2}}. \tag{61}$$

Contextual review is applied only after the corresponding numerical gate has been satisfied. It determines whether a numerically qualified candidate is retained or excluded, but never introduces tokens outside the gated candidate pools.

C.3 Derivation of Token-Specific Logit Correction

For a target token k, its probability under the softmax function can be written as:

$$p(k) = \frac{e^{z_k}}{e^{z_k} + \sum_{j \neq k} e^{z_j}}, \tag{62}$$

where z_k denotes the logit of token k. Let

$$S = \sum_{j \neq k} e^{z_j}, \tag{63}$$

then the probability of token k can be rewritten as:

$$p(k) = \frac{e^{z_k}}{e^{z_k} + S}. \tag{64}$$

The corresponding logit transformation is defined as:

$$\text{logit}(p(k)) = \log \frac{p(k)}{1 - p(k)}. \tag{65}$$

Substituting the probability expression into the logit transformation gives:

$$\text{logit}(p(k)) = \log \frac{\frac{e^{z_k}}{e^{z_k}+S}}{\frac{S}{e^{z_k}+S}} = \log \frac{e^{z_k}}{S} = z_k - \log S. \tag{66}$$

Under the independent token calibration assumption, only the target token logit is adjusted while all other quantized logits remain unchanged. Let $z'_{Q,k}$ denote the corrected logit whose probability matches the full-precision preference. The required correction magnitude is therefore:

$$c_{Q,i}(k) = z_{Q,k} - z'_{Q,k} = \text{logit}(p_Q(k|x_{<t_i})) - \text{logit}(p_F(k|x_{<t_i})). \tag{67}$$

D Dialogue Examples

Table 6 shows how RATIO curbs repetition and erroneous continuations while preserving correct answers.

Table 6: Dialogue examples on Qwen-1.5B with GPTQ-W3. Red marks repetitive or erroneous reasoning; green marks correct RATIO calculations and answers. Ellipses indicate omissions.

Case 1: Repetitive reasoning after reaching the correct result	
Prompt	From her science class study, Brady learned that each whale has 40 gallons of blood and that a shark has three times as much blood as a whale. How many gallons of blood do ten sharks have?
GPTQ-W3	Each shark has $3 \times 40 = 120$ gallons, so ten sharks have $10 \times 120 = 1200$ gallons. Wait, let me double-check that. ... Wait, that seems correct. Let me think if I interpreted the question correctly. ... Alternatively, perhaps I can think of the problem as ten sharks each having three times as much as a whale, so $10 \times (3 \times W) = 3 \times (10 \times W)$. Therefore, ten sharks have $3 \times 10 \times 40 = 1200$ gallons. Alternatively, perhaps I can think of it as $10 \times (3 \times W) = 3 \times (10 \times W)$. ... Wait, 3 W. ... 3 W3 W3 W.
RATIO	A shark has three times as much blood as a whale. Therefore, each shark has $3 \times 40 = 120$ gallons of blood. Ten such sharks have $10 \times 120 = 1200$ gallons of blood. The final answer is $\boxed{1200}$.
Case 2: Repetition followed by an incorrect reasoning trajectory	
Prompt	Compute $99^2 + 99 + 1$ in your head.
GPTQ-W3	First, $99^2 = (100 - 1)^2 = 10{,}000 - 200 + 1 = 9{,}801$. Adding 99 gives 9,900, and adding 1 gives 9,901. ... Let me verify each step again. 99^2 is 9,801, then $9{,}801 + 99 = 9{,}900$, and adding 1 gives 9,901. ... Alternatively, maybe I can factor it as $99 \times (99 + 1) + 1$. That gives $99 \times 100 + 1 = 9{,}901$. Alternatively, if I think of $99^2 + 99 + 1$ as $(99 + 1)^2 - 1$, that is $100^2 - 1 = 9{,}999$. Wait, no, that is incorrect. ... So, 9999. ... 10999999999999999999999999999999999999999999.
RATIO	First, $99^2 = (100 - 1)^2 = 10{,}000 - 200 + 1 = 9{,}801$. Next, $9{,}801 + 99 = 9{,}900$, and $9{,}900 + 1 = 9{,}901$. Thus, the final answer is $\boxed{9901}$.