LAQuant: A Simple Online Transform-free Large Reasoning Model Quantization by Layer-wise Lookahead Loss

Euntae Choi[*]
Seoul National University
euntae.choi175@gmail.com

Sumin Song[*]
Seoul National University
songsm921@snu.ac.kr

Sungjoo Yoo[†]
Seoul National University
sungjoo.yoo@gmail.com

Abstract

Large reasoning models (LRMs) reach competition-level math and coding accuracy via long autoregressive decoding, making per-token decoding cost a primary deployment concern. Weight quantization is the standard tool for acceleration, but representative recipes — including state-of-the-art end-to-end (E2E) quantization-aware training (QAT) — lose accuracy on long-decoding reasoning benchmarks despite preserving perplexity and short-decode accuracy. Through a systematic gradient-direction analysis, we identify two factors driving this gap: (i) KV-cache fidelity preservation under the QAT loss, which E2E supervision attenuates via the softmax Fisher metric; and (ii) Hessian-subspace alignment between calibration data and the deployment distribution. We propose LookAhead Quantization (LAQuant), a layer-wise weight-only QAT method that addresses both factors without online-transform overhead by combining reasoning-domain calibration with a one-layer lookahead loss whose implicit cross-layer co-adaptation preserves the next-layer residual stream. For Qwen3-4B under W3G128 quantization, LAQuant improves AIME25 Pass@1 over ParoQuant by 15.11 pp (1.93 pp over ParoQuant++ at matched calibration) while achieving a 3.42× decoding speedup over FP16 on RTX A6000, compared with ParoQuant's 3.01×. Code is made available at `https://github.com/songsm921/LAQuant`.

1 Introduction

Large reasoning models (LRMs) [59, 50, 18], powered by reinforcement learning [50, 60] and test-time scaling [51], have recently achieved strong performance on challenging STEM and coding tasks. Since accuracy scales with reasoning-trace length, per-token decoding efficiency is a critical deployment concern.

Quantization is a common and essential technique for efficient LLM inference [63, 36, 2, 13, 35, 49, 52, 38], and its importance is amplified under the long-decoding workloads of LRMs. For conventional LLMs, quantization algorithms have advanced significantly: some methods push weight-only quantization to 2-bit [10, 4, 3, 38] or even binary [21], while others jointly quantize weights, activations, and the KV cache at 4-bit [2, 39]. On competitive reasoning benchmarks [25, 42, 48, 24] that require long autoregressive decoding, however, representative methods such as AWQ [35], GPTQ [13], and EfficientQAT [4] are reported to exhibit visible degradation even under 4-bit weight-only quantization, indicating that success on conventional benchmarks does not transfer to long-decode reasoning settings.

[*]Equal contribution.
[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In response, recent works have proposed quantization methods specifically targeting LRMs. ParoQuant [33] combines minimal pairwise rotation with channel scaling to reduce per-layer quantization error, but its non-absorbable rotation introduces per-token inference overhead, which is non-trivial given the long thinking-token budgets typical of LRM deployment. ReasoningQAT [46] chains layer-wise quantization-aware training (QAT) with end-to-end (E2E) QAT under a domain-mixed calibration corpus, improving over post-training methods but still leaving headroom on competition-level benchmarks. Despite these advances, the question of why LRM quantization is more difficult than its conventional counterpart remains open.

The E2E QAT objective optimizes only the final logit distribution, placing no direct constraint on per-layer KV-cache errors — yet KV-cache fidelity is precisely what long-decoding quality requires, since each generated token re-attends to every prior K and V projection. A gradient-direction analysis identifies two factors driving this difficulty: (i) KV-cache fidelity preservation under the QAT loss, and (ii) Hessian-subspace alignment between calibration data and deployment. We propose **L**ook**A**head **Quant**ization (LAQuant), a layer-wise weight-only QAT addressing both with zero online-transform overhead. We focus on W3G128, where per-layer error and KV-cache compounding are most severe; the analysis and method extend to W4G128. LAQuant pairs a one-layer lookahead loss with a Hessian-aligned calibration corpus: the lookahead loss is an implicit Jacobian-weighted error attribution that couples the seven trainable projections of each layer through the next-layer output, producing cross-layer co-adaptation; the calibration corpus keeps the optimization target faithful to the deployment subspace. Our contributions are summarized as follows:

- **A gradient-direction framework** for LRM weight-only QAT that identifies two factors driving the quantization gap: end-to-end supervision attenuates per-layer learning signals (including for KV-cache directions) via the softmax Fisher metric, and calibration-data choice is governed by activation-Hessian subspace alignment with the deployment distribution.
- **LAQuant**, a unified layer-wise QAT recipe that combines lookahead-based loss reshaping with Hessian-aligned calibration data to jointly address both factors, without any non-absorbable online transformation.
- **State-of-the-art empirical results** on challenging STEM reasoning benchmarks, with complementary improvements over the two leading paradigms: a **19.38 pp** AIME25 Pass@1 gain over ReasoningQAT on Qwen3-4B W3G128, and a **3.42**× decoding speedup over FP16 on RTX A6000 GPU (vs ParoQuant's 3.01×), establishing a new accuracy–speed Pareto frontier for weight-only quantization of large reasoning models.

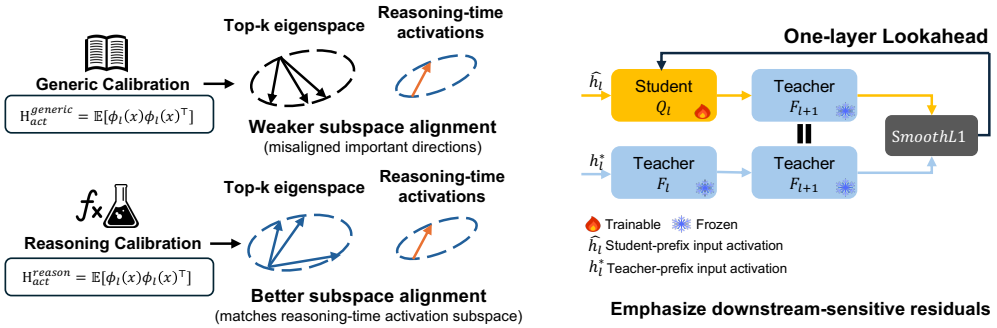
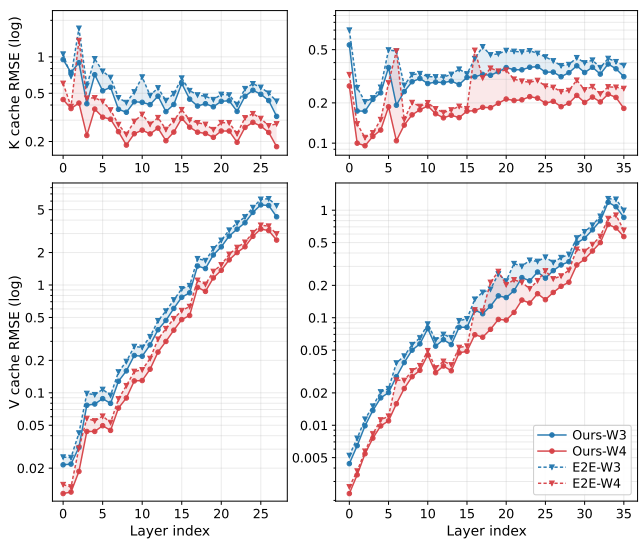


Figure 1: Overview of **LAQuant**. We combine a one-layer lookahead loss (Section 4.4) with reasoning-domain calibration data (Section 4.1) in a layer-wise QAT pipeline that produces standard weight-only quantized models with no online transformations.

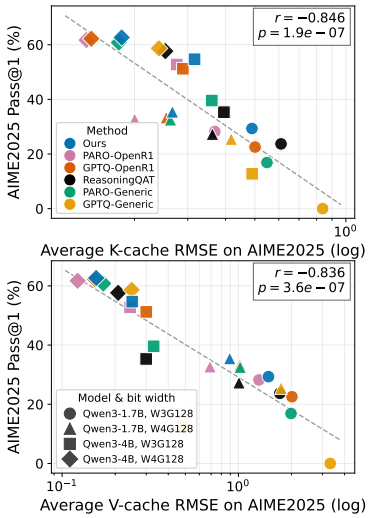
2 Preliminaries

LLM Weight Quantization. Linear weight quantization maps a full precision weight w to b-bit integer indices q, scales s, and zeros z as follows:

$$q \;=\; \mathrm{clip}\big(\lfloor \tfrac{w}{s} \rceil + z,\; 0,\; 2^b - 1\big)\,, \qquad \hat{w} \;=\; s \cdot (q - z), \tag{1}$$



(a) Per-layer K- and V-cache RMSE on AIME25 reasoning traces.



(b) K-cache (top) and V-cache (bottom) RMSE vs. AIME25 Pass@1 across 24 quantized checkpoints.

Figure 2: KV-cache fidelity is strongly associated with reasoning accuracy. (a) End-to-end QAT (ReasoningQAT) sacrifices both K- and V-cache fidelity across layers, while LAQuant preserves both. Top row: K-cache RMSE; bottom row: V-cache RMSE. (b) Across 24 quantized Qwen3-1.7B/4B checkpoints spanning 6 methods at W3G128 and W4G128, K- and V-cache RMSE on AIME25 traces both strongly anti-correlate with downstream AIME25 Pass@1; the two cache metrics are themselves highly correlated, consistent with a shared KV-cache fidelity factor.

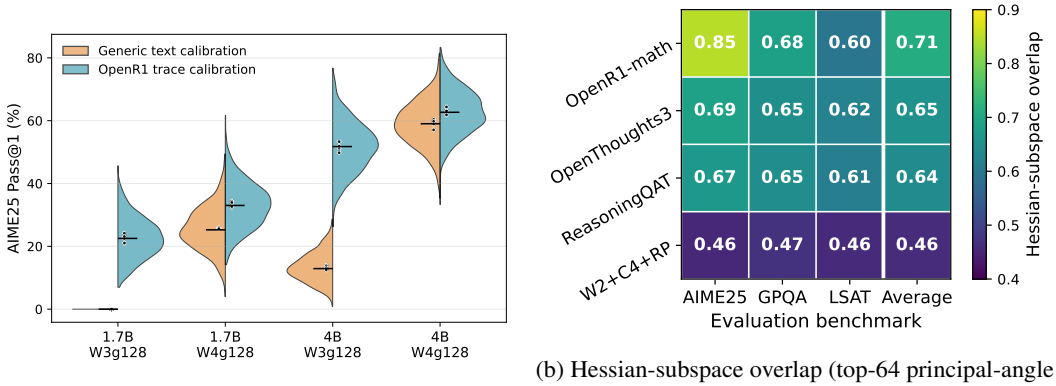
where $\lfloor \cdot \rceil$ denotes round-to-nearest and $\hat{w}$ is the dequantized weight. Grouped quantization computes the scales and zeros on each consecutive g elements in the rows of the weight, which is denoted WbGg (e.g., W3G128 for 3-bit weights with group size 128).

Layer-wise QAT. Layer-wise QAT [4] optimizes the transformer block-by-block, minimizing the reconstruction error between the FP16 teacher's layer output and the quantized student's layer output. The learnable parameters are the per-layer quantization parameters (scales and zeros) and optionally the weights, clipping bounds [49], and channel scales [58, 35]; gradients pass through the rounding operation via the straight-through estimator. A full related-work survey appears in Appendix A.

3 Motivation

KV-Cache Quality is Critical for Robust LRM Quantization. Recent studies [32, 37, 64, 43, 46, 33] report that legacy PTQ and QAT algorithms degrade on long-decoding reasoning tasks such as AIME [25, 42] (competition math) and LiveCodeBench [24] (code generation), which generate up to 32k tokens per response [50, 59, 18, 51, 45]. Weight quantization induces errors in the output activations of the key and value projections, which populate the KV cache attended by every subsequent token. Because each generated token attends to all prior keys and values, both K-cache errors (corrupting the attention routing through $\mathrm{softmax}(QK^\top/\sqrt{d})$) and V-cache errors (corrupting the softmax-weighted output mixture) are consequential: they compound along two axes, namely (i) later transformer layers receive increasingly distorted inputs, and (ii) every future token must attend to a degraded KV-cache throughout decoding.

In contrast, conventional LLM quantization [13, 35, 10, 4, 49, 58, 2, 39, 30] is primarily evaluated on perplexity, zero-shot accuracy, and short-decode tasks, where KV-cache errors have little opportunity to compound along the sequence axis and output-level supervision (SFT loss or KL divergence against a BF16 teacher) proves highly effective; recent advances have accordingly moved toward end-to-end QAT [10, 4, 39, 30].



(a) AIME25 Pass@1 with generic vs. reasoning calibration.

(b) Hessian-subspace overlap (top-64 principal-angle cosine) between calibration corpora and evaluation traces on Qwen3-1.7B.

Figure 3: Calibration data choice substantially affects reasoning accuracy. (a) Under the same GPTQ algorithm, switching the calibration corpus from a generic pretraining-text mix to DeepSeek-R1 reasoning traces yields large Pass@1 gains. (b) A plausible mechanism: reasoning-domain calibration corpora (OpenR1-math, OpenThoughts3, ReasoningQAT mix) induce a Hessian subspace that overlaps far more with the inference distribution than a generic-text mix (WikiText2+C4+RedPajama); the calibration ranking is invariant under $k \in [4, 256]$ (Appendix E).

A natural hypothesis is that this logit-matching objective should carry over to long-decoding tasks. Our analysis (Figure 2a) refutes this: a recent end-to-end QAT model [46] sacrifices both K- and V-cache fidelity across layers to match the BF16 teacher's logits under teacher-forcing, a trade that collapses under long autoregressive decoding where each generated token re-attends to the damaged KV cache. Our layer-wise model instead aligns intermediate hidden representations, achieving $1.25\times$ lower aggregate K-cache and $1.16\times$ lower aggregate V-cache RMSE than the E2E baseline, and recovers the reasoning accuracy quantified in Section 5.1. The relationship is general: across 24 quantized checkpoints spanning 6 methods at W3G128 and W4G128, K- and V-cache RMSE (per-token average) on AIME25 traces both strongly anti-correlate with downstream Pass@1 (Pearson $r \approx -0.85$ for both), and are themselves highly correlated ($r = +0.85$), consistent with a shared KV-cache fidelity factor (Figure 2b).

LRM Quantization is Sensitive to Calibration Data, Especially under Challenging Scenarios. Existing LLM quantization algorithms typically calibrate on pre-training text corpora (e.g., WikiText2 [44], C4 [47], RedPajama [55]), and the generalization capability of modern LLMs has made this choice sufficient for most tasks. For LRMs, however, calibration data choice is a first-order factor. Figure 3a compares GPTQ-quantized LRMs under two calibration sources: a *generic* mixture of pre-training text, and a *reasoning* corpus of DeepSeek-R1 traces on OpenR1-Math-220k problems. Reasoning calibration yields up to 38.45 pp improvement on AIME2025 Pass@1 (Qwen3-4B at W3G128 under GPTQ; Table 13), with the largest gains at smaller model scales and lower bit widths — e.g., on Qwen3-1.7B at W3G128, switching corpora moves GPTQ from 0.00% to 22.55% AIME25 Pass@1 (Appendix G.1). Prior works [46, 56] have reported similar observations but only empirically; we provide a principled explanation based on Hessian-subspace alignment in Section 4.1.

Runtime-Overhead-Free Quantization Is Preferable. Rotation-based techniques [2, 39, 33, 34] reduce per-layer quantization error, but every known formulation includes non-absorbable rotations applied online at every forward pass; even with efficient Hadamard kernels [7, 1, 36], the added per-token latency dominates total cost in long-decoding LRM inference. Online rotations also complicate deployment process: serving frameworks target general CUDA-core kernels and rarely include Hadamard primitives, and newer rotation variants [39, 34] require custom full-matrix transforms. These considerations motivate LAQuant, which produces standard integer weights with per-group scales and zeros and deploys on any existing weight-only quantized kernel.

4 LAQuant: Layer-Wise QAT with One-Layer Lookahead

We justify the design of LAQuant through two analytical strands, one for each factor identified in Section 3: (a) calibration-data alignment via activation Hessians (Section 4.1), and (b) a gradient-direction analysis (Section 4.2) of supervision level (Section 4.3) and loss shape (Section 4.4).

4.1 Hessian-Subspace Analysis of Calibration Data

A widespread empirical observation [37, 41, 46, 56, 57] is that LRMs calibrated on reasoning traces substantially outperform those calibrated on generic text. We attribute this gap to an activation–Hessian subspace-alignment phenomenon, formalized as follows. For any calibration distribution $\mathcal{D}_{\mathrm{cal}}$, let $x \sim \mathcal{D}_{\mathrm{cal}}$ denote a calibration token sequence, and let $\phi_\ell(x) \in \mathbb{R}^d$ denote the layer-ℓ input activation induced by x. The linearized layer-wise reconstruction objective with respect to the layer weight matrix $W \in \mathbb{R}^{d_{\mathrm{out}} \times d}$ expands, near its unconstrained optimum $W^\star$, as

$$\mathcal{L}_{\mathrm{cal}}(W) \;\approx\; \mathrm{Tr}\big[(W - W^\star)\, H_{\mathrm{act}}^{\mathrm{cal}}\, (W - W^\star)^\top\big], \qquad H_{\mathrm{act}}^{\mathrm{cal}} \triangleq \mathbb{E}_{x \sim \mathcal{D}_{\mathrm{cal}}}\big[\phi_\ell(x)\phi_\ell(x)^\top\big], \quad (2)$$

so the curvature is governed by the second-order activation statistics of the calibration distribution. Under a finite bit budget, the trained $\hat{W}$ allocates precision to high-curvature eigendirections of $H_{\mathrm{act}}^{\mathrm{cal}}$ and leaves low-curvature directions weakly constrained [12, 29, 20, 13]; if the deployment-activation distribution populates a different subspace, $\hat{W}$ is systematically misallocated at inference.

The natural diagnostic for this mismatch is the average top-k principal-angle cosine $\bar{c}_k(\mathcal{D}_{\mathrm{cal}}, \mathcal{D}_{\mathrm{eval}})$ between the top-k eigenspaces of $H_{\mathrm{act}}^{\mathrm{cal}}$ and $H_{\mathrm{act}}^{\mathrm{eval}}$. LAQuant adopts a reasoning-trace calibration corpus (DeepSeek-R1 traces over OpenR1-Math-220k [23]); its alignment with deployment activations is reported in Figure 3b.

4.2 Gradient-Direction Framework

Definition 1 (Per-layer quantization residual) *For a quantized layer Q_ℓ with learnable parameters $\theta_q^{(\ell)}$, trained against teacher layer F_ℓ, the per-layer quantization residual is*

$$\varepsilon_\ell(\theta_q^{(\ell)}) \;\triangleq\; Q_\ell\big(\hat{h}_\ell;\, \theta_q^{(\ell)}\big) - F_\ell(h_\ell^\star), \quad (3)$$

where $\hat{h}_\ell$ and $h_\ell^\star$ are the student-prefix and teacher-prefix layer-ℓ inputs respectively, with preceding student layers frozen and quantized.

The residual is a high-dimensional vector in $\mathbb{R}^{B \cdot T \cdot d_{\mathrm{hidden}}}$, whose downstream effect depends not only on its magnitude but also on its direction. Under a first-order linearization, components seen by sensitive downstream projections, such as the K-projection in layer $\ell + 1$, can perturb the attention keys and affect future-token attention routing. By contrast, components in directions weakly observed by the downstream Jacobian have limited first-order impact.

For any QAT loss $\mathcal{L}$ admitting a quadratic Taylor expansion in ε_ℓ near the unconstrained optimum, the gradient takes the form

$$\nabla_{\theta_q^{(\ell)}} \mathcal{L} \;=\; J_\theta^\top\, M\, \varepsilon_\ell \;+\; O\big(\|\varepsilon_\ell\|^2\big), \qquad J_\theta \triangleq \partial \varepsilon_\ell / \partial \theta_q^{(\ell)}, \quad (4)$$

where $M \succeq 0$ is a *local* metric on the residual space, determined by the choice of loss together with the computation graph through which the residual is evaluated and the linearization point. The metric M specifies how the residual's directional components are weighted on their way into the gradient; the next two subsections instantiate Eqn. (4) for LAQuant's two loss-axis design decisions.

4.3 Supervision Level: Layer-Wise over End-to-End

Proposition 1 (Loss-induced metrics for E2E and layer-wise QAT; proof in Appendix B) *Under the local Gauss–Newton approximation around $\varepsilon_\ell = 0$, end-to-end QAT against the teacher's softmax with forward-KL loss yields*

$$M_{\mathrm{E2E}} \;=\; \mathcal{J}^\top\, H_{\mathrm{KL}}\, \mathcal{J}, \qquad H_{\mathrm{KL}} \;=\; \mathrm{diag}(p^\star) - p^\star (p^\star)^\top, \quad (5)$$

where $\mathcal{J}$ maps the layer-ℓ residual to the pre-softmax logits through the remaining $L-\ell$ quantized layers, the final RMSNorm, and the language-model head, and $p^\star$ is the teacher's softmax distribution at the position. The layer-wise reconstruction loss (squared L_2 or its locally quadratic variants such as SmoothL1) yields, exactly,

$$M_{\mathrm{lw}} \;=\; I. \tag{6}$$

The matrix H_{KL} is the Fisher information of the teacher's categorical distribution, with spectral norm bounded by $\sigma_{\max}(H_{\mathrm{KL}}) \le 2\max_i p_i^\star(1-p_i^\star) \le 1/2$ (vanishing as $\max_i p_i^\star \to 1$; see Appendix B); M_{E2E} is therefore the softmax Fisher metric carried through $\mathcal{J}$ to the residual.

Combining Proposition 1 with this spectral bound gives $\|M_{\mathrm{E2E}}\, v\| \le 2\max_i p_i^\star(1-p_i^\star)\cdot\sigma_{\max}(\mathcal{J})^2 \cdot \|v\|$ for every residual direction v, whereas $\|M_{\mathrm{lw}}\, v\| = \|v\|$ (Proposition 3, Appendix B). Beyond the operator-norm bound, M_{E2E} is strongly anisotropic in practice on AIME25 traces: residual perturbations along its leading eigendirection are substantially more responsive than along random directions (Appendix C). This anisotropy cannot be removed by any scalar learning-rate schedule or global loss reweighting since it is a structural property of the E2E gradient.

The KV-cache-reducing residual directions $g_K \triangleq \nabla_{h_\ell}\|\Delta K_{\ell+1}\|^2$ (and analogously g_V for V), decisive for long-decoding reasoning (Section 3), sit in the small-eigenvalue subspace of M_{E2E} rather than the leading subspace E2E supervision actually trains, with relative responsiveness comparable to a random-Gaussian baseline (Appendix C). LW supervision, with $M_{\mathrm{lw}} = I$, treats every residual direction equally and so retains the K/V learning signal. LAQuant therefore adopts pure layer-wise supervision (Appendix D confirms via a controlled E2E baseline); the lookahead loss in Section 4.4 contributes a complementary mechanism, cross-layer co-adaptation, analyzed next.

4.4 Loss Shape: One-Layer Lookahead

We refine the layer-wise objective with a one-layer lookahead (LA) loss

$$\mathcal{L}_{\mathrm{la}}(\theta_q^{(\ell)}) \;\triangleq\; \mathrm{SmoothL1}\Big(F_{\ell+1}\big(Q_\ell(\hat{h}_\ell;\,\theta_q^{(\ell)})\big),\; F_{\ell+1}\big(F_\ell(h_\ell^\star)\big)\Big), \tag{7}$$

which propagates the residual through the next teacher layer before measuring error. At the final layer $\ell = L$, $F_{\ell+1}$ does not exist; we set $\mathcal{L}_{\mathrm{la}}$ to the layer-wise reconstruction loss for that layer.

Proposition 2 (Lookahead loss-induced metric (local first-order); proof in Appendix B)
Under a first-order expansion of $F_{\ell+1}$ around $F_\ell(h_\ell^\star)$, the lookahead loss (7) *induces the local Gauss–Newton metric*

$$M_{\mathrm{la}} \;=\; J_{\ell+1}^\top J_{\ell+1}, \qquad J_{\ell+1} \triangleq \partial F_{\ell+1}/\partial h\,\big|_{h=F_\ell(h_\ell^\star)}. \tag{8}$$

Unlike M_{E2E} from Proposition 1, M_{la} contains no vocabulary-softmax factor and no composition of multiple downstream Jacobians.

M_{la} inherits the *Jacobian-weighted attribution* property of $M_{\mathrm{E2E}} = \mathcal{J}^\top H_{\mathrm{KL}}\mathcal{J}$: both reweight residual directions by the sensitivity of downstream computation to the residual. Unlike M_{E2E}, however, M_{la} contains neither the softmax Fisher factor H_{KL} nor the composition of $L-\ell$ downstream Jacobians, both of which are responsible for E2E gradient dilution (Proposition 3). Lookahead supervision therefore retains the structural advantage of E2E (knowing where the residual will land in the next-layer computation) while avoiding the spectral attenuation that dilutes E2E gradients along the very residual directions that long-decoding reasoning needs.

Because M_{la} is shaped by layer $\ell{+}1$'s Jacobian rather than by layer ℓ's alone, the optimization of layer ℓ is informed by how layer $\ell{+}1$ will use the residual: the optimizer favors configurations whose projection errors *combine* into a small next-layer residual rather than minimizing any single projection's marginal error. We call this property *cross-layer co-adaptation*. The coupled projection set includes K and V, so lookahead reduces K- and V-projection output RMSE alongside the other five projections (Table 5b), directly improving the KV-cache fidelity that Section 3 identifies as critical for long-decoding accuracy. Table 5a further reports performance as the lookahead depth varies.

5 Evaluation

Models and Benchmarks. We apply LAQuant on Llama-3.1 8B [16], DeepSeek-R1-distilled Llama-3.1 8B [8], and Qwen3 (Base + chat variants at 1.7B, 4B, 8B) [59]. Following ParoQuant [33],

we evaluate LAQuant on three different types of benchmarks: i) perplexity on WikiText2 and C4, ii) zero-shot accuracy on ARC-Challenge, ARC-Easy [6], BoolQ [5], and HellaSwag [61], and iii) reasoning accuracy on AIME2024 [25], AIME2025 [42], GPQA Diamond [48], LSAT-AR [65], MMLU-Pro [54], and LiveCodeBench [24] (2024.10-2025.02 subset as used in the official Qwen3 evaluation[3]).

Table 1: Pass@1$^{\uparrow}$ on **reasoning tasks** under 3- and 4-bit quantization with a group size of 128. R-QAT and ParoQ denote ReasoningQAT and ParoQuant, respectively; ParoQ++ uses the same reasoning-domain dataset and training data size as LAQuant for a controlled experiment, and R-QAT uses its own data selection heuristic. Detailed results are reported in Appendix G.1.

(a) W3G128 Pass@1 on reasoning tasks.

Model	Method	AIME24	AIME25	GPQA	LSAT	MMLU-Pro	LCB	Avg.	Spd.
Qwen3-1.7B	FP16	45.84	35.94	40.70	64.10	57.31	33.71	46.27	1.0×
	GPTQ	0.00	0.00	25.03	18.78	22.34	0.00	11.03	2.29×
	R-QAT	25.52	23.70	31.03	47.69	47.48	19.90	32.55	2.29×
	ParoQ	7.24	16.88	30.49	36.10	43.79	12.94	24.57	2.05×
	ParoQ++	33.65	28.23	30.81	49.30	45.84	19.40	34.54	2.05×
	LAQuant	**34.64**	**29.33**	**31.79**	**54.86**	**47.94**	**21.02**	**36.60**	2.29×
Qwen3-4B	FP16	73.23	63.75	55.59	81.14	70.23	54.48	66.40	1.0×
	GPTQ	12.92	12.76	33.65	40.41	53.32	2.99	26.01	3.42×
	R-QAT	43.75	35.31	**46.21**	67.64	**63.24**	37.31	48.91	3.42×
	ParoQ	41.30	39.58	44.41	68.94	63.41	38.31	49.33	3.01×
	ParoQ++	59.74	52.76	46.17	77.07	62.41	45.03	57.20	3.01×
	LAQuant	**62.92**	**54.69**	**46.21**	**77.15**	62.63	**45.15**	**58.12**	3.42×
Qwen3-8B	FP16	74.95	67.66	58.78	85.57	73.44	58.46	69.81	1.0×
	GPTQ	44.48	40.68	48.17	69.92	65.71	37.94	51.15	5.15×
	R-QAT	53.75	43.91	48.61	77.23	68.33	44.16	56.00	5.15×
	ParoQ	59.53	52.92	52.30	78.53	**68.93**	44.65	59.48	4.63×
	ParoQ++	67.19	58.39	53.09	79.48	63.32	51.12	62.10	4.63×
	LAQuant	**68.33**	**61.30**	**53.66**	**83.80**	66.86	**51.74**	**64.28**	5.15×
R1-Distill-Llama-8B	FP16	43.44	31.20	48.17	59.78	58.84	36.32	46.29	1.0×
	GPTQ	15.45	19.64	37.97	42.47	47.10	24.13	31.13	5.56×
	R-QAT	30.16	27.50	40.50	48.48	51.05	29.11	37.80	5.56×
	ParoQ	0.89	0.05	32.99	29.65	44.29	1.12	18.17	4.90×
	ParoQ++	30.63	25.57	40.37	48.89	51.44	29.47	37.73	4.90×
	LAQuant	**31.46**	**28.54**	**43.31**	**50.98**	**51.70**	**31.22**	**39.54**	5.56×

(b) W4G128 Pass@1 on reasoning tasks.

Model	Method	AIME24	AIME25	GPQA	LSAT	MMLU-Pro	LCB	Avg.	Spd.
Qwen3-1.7B	FP16	45.84	35.94	40.70	64.10	57.31	33.71	46.27	1.0×
	GPTQ	24.17	25.47	33.14	48.29	50.49	20.40	33.66	1.64×
	R-QAT	32.61	27.29	35.70	53.48	54.44	25.87	38.23	1.64×
	ParoQ	39.53	32.55	35.86	58.30	54.58	28.61	41.57	1.48×
	ParoQ++	41.82	32.71	36.30	60.52	55.44	28.98	42.63	1.48×
	LAQuant	**42.35**	**35.47**	**36.43**	**61.14**	**55.58**	**31.84**	**43.80**	1.64×
Qwen3-4B	FP16	73.23	63.75	55.59	81.14	70.23	54.48	66.40	1.0×
	GPTQ	63.44	58.67	51.61	78.10	68.03	48.75	61.43	2.0×
	R-QAT	64.64	57.66	52.15	78.45	68.28	46.76	61.32	2.0×
	ParoQ	67.29	60.78	52.46	79.43	**69.41**	50.00	63.23	1.84×
	ParoQ++	70.37	61.72	53.44	80.87	68.73	51.62	64.46	1.84×
	LAQuant	**70.42**	**62.66**	**53.60**	**81.03**	68.41	**52.74**	**64.81**	2.0×
Qwen3-8B	FP16	74.95	67.66	58.78	85.57	73.44	58.46	69.81	1.0×
	GPTQ	71.56	65.21	57.89	82.31	**72.67**	52.99	67.10	2.45×
	R-QAT	68.44	59.48	**58.49**	**85.24**	72.65	51.62	65.99	2.45×
	ParoQ	73.39	64.52	56.85	84.43	72.43	55.47	67.85	2.32×
	ParoQ++	73.81	64.27	56.66	83.99	72.59	55.72	67.84	2.32×
	LAQuant	**74.17**	**66.98**	57.23	**85.24**	71.52	**56.69**	**68.62**	2.45×
R1-Distill-Llama-8B	FP16	43.44	31.20	48.17	59.78	58.84	36.32	46.29	1.0×
	GPTQ	38.44	29.85	**46.65**	56.66	**56.78**	34.21	43.66	2.46×
	R-QAT	40.63	29.53	46.09	57.01	55.96	34.33	43.92	2.46×
	ParoQ	39.12	28.54	46.34	54.81	56.69	33.33	43.14	2.33×
	ParoQ++	38.86	27.60	46.46	56.66	56.77	35.45	43.63	2.33×
	LAQuant	**41.41**	**31.77**	45.55	**58.67**	54.98	**35.70**	**44.68**	2.46×

[3] `https://qwen.ai/blog?id=qwen3`

Implementation Details. We use group-128 weight-only quantization at 3 and 4 bits. For reasoning evaluation, QAT data is the first 4,096 samples of OpenR1-Math-220k [23] (verified disjoint from our benchmarks); we apply the Qwen-style chat template to prompts and DeepSeek-R1 responses and right-truncate to 2,048 tokens including formatting. For fair comparison, GPTQ is calibrated using the same data budget as the QAT-based methods in each setting. For base models on general tasks, we use a WikiText2/C4/RedPajama mixture.

Our framework builds on the official EfficientQAT [4] implementation, training each layer for 20 epochs with embedding and LM head frozen. Weights, quantization parameters (scales and zeros), learnable clipping bounds [49], and per-channel scaling [58] are jointly trained with AdamW [40]. Following ParoQuant [33], we use SmoothL1 [15, 22] as the reconstruction loss. The learning rate for weights is set to half that of the other trainable quantities (2.5×10^{-5}/5×10^{-5} for Llama, 5×10^{-5}/1×10^{-4} for Qwen3).

5.1 Accuracy Results

Reasoning Tasks. Table 1 reports Pass@1 on six reasoning tasks under 4-bit and 3-bit quantization. The results first show that reasoning-domain calibration data are crucial for preserving reasoning ability in the low-bit regime. Under W3G128, ParoQuant++ consistently outperforms ParoQuant across the models, with the average accuracy on Qwen3-1.7B improving from 24.57 to 34.54, corresponding to a **40.6%** relative gain. The effect is most pronounced on R1-Distill-Llama-8B at W3G128, where ParoQuant's default calibration collapses AIME25 Pass@1 to 0.05 while ParoQuant++ recovers it to 25.57 — a $\sim$20 pp six-task average gain ($18.17 \rightarrow 37.73$) attributable to calibration choice alone.

The results also highlight the advantage of lookahead quantization as an efficient alternative to online transformations. Compared with ParoQuant++, our method further improves reasoning performance while eliminating inference overhead. On DeepSeek-R1-Distill-Llama-8B, LAQuant improves the average accuracy from 43.63 to 44.68, corresponding to a **2.4%** relative gain, while also increasing decoding speedup from 2.33× to 2.46×. This shows that lookahead quantization can provide additional reasoning performance without incurring the inference overhead of online transformations. Detailed results for each reasoning benchmark are provided in Appendix G.1.

Table 2: Results on **general tasks** under 4- and 3-bit quantization. Perplexity is evaluated at a context length of 8192. 0-shot denotes the average accuracy across ARC-C, ARC-E, BoolQ, and HellaSwag.

Method	Bits	Qwen3-1.7B-Base			Qwen3-4B-Base			Qwen3-8B-Base			Llama3.1-8B		
		WikiText2$^{\downarrow}$	C4$^{\downarrow}$	0-shot$^{\uparrow}$	WikiText2$^{\downarrow}$	C4$^{\downarrow}$	0-shot$^{\uparrow}$	WikiText2$^{\downarrow}$	C4$^{\downarrow}$	0-shot$^{\uparrow}$	WikiText2$^{\downarrow}$	C4$^{\downarrow}$	0-shot$^{\uparrow}$
FP16	16	8.31	8.71	59.95	7.01	7.63	66.28	6.23	6.97	68.95	5.61	6.76	68.68
GPTQ	3	10.56	10.88	51.72	8.12	8.71	60.11	6.96	7.65	67.34	6.79	7.90	63.08
ParoQ	3	**8.91**	**9.38**	58.53	7.57	8.21	**66.01**	**6.56**	**7.32**	68.61	6.38	7.49	65.51
LAQuant	3	8.99	9.52	**58.61**	**7.44**	**8.16**	65.39	6.57	7.37	**69.35**	**6.31**	**7.48**	**65.64**
GPTQ	4	8.74	9.14	58.95	7.23	7.85	65.79	6.39	7.11	**69.48**	5.85	6.98	68.25
ParoQ	4	**8.43**	**8.83**	**60.75**	**7.09**	**7.73**	**67.36**	**6.29**	**7.04**	69.32	**5.78**	**6.91**	68.12
LAQuant	4	8.46	8.91	60.20	7.12	7.77	66.44	6.32	7.07	69.32	5.80	6.94	**68.36**

General Tasks. On general tasks (Table 2), LAQuant matches ParoQuant on perplexity and zero-shot accuracy across all evaluated models in both W4G128 and W3G128, while avoiding online transformations. Per-task results are provided in Appendix G.2.

5.2 Efficiency Results

Table 3: Decoding (batch size = 1) throughput (tokens/sec)$^{\uparrow}$ on RTX A6000.

RTX A6000	Bits	Qwen3-1.7B	Qwen3-4B	Qwen3-8B	Llama3.1-8B
FP16	16	138.4	67.6	38.2	40.47
ParoQuant	4	205.3 (x1.48)	124.5 (x1.84)	88.7 (x2.32)	94.35 (x2.33)
LAQuant	4	**227.3** (x1.64)	**135.1** (x2.00)	**93.8** (x2.45)	**99.74** (x2.46)
ParoQuant	3	283.7 (x2.05)	203.5 (x3.01)	177.0 (x4.63)	198.13 (x4.90)
LAQuant	3	**316.9** (x2.29)	**231.3** (x3.42)	**196.8** (x5.15)	**225.01** (x5.56)

Inference. Table 3 shows decoding throughput on an RTX A6000. For INT3, we adopt the GPTQ GEMV kernel with additional tuning for smaller model sizes, since the original kernel was designed for 175B-scale models; we describe the modifications in detail in Appendix H. To ensure a fair

comparison, we integrate our INT3 kernel into the ParoQuant benchmark pipeline (`StaticCache` + `torch.compile` + CUDA graphs) and run both with and without the online rotation. For INT4, both LAQuant and ParoQuant use the same AWQ GEMV kernel; the only difference is whether the rotation is applied.

Removing the online rotation consistently improves throughput: 6–11% over ParoQuant at W4 (avg 7.7%) and 11–14% at W3 (avg 12.5%). The W3 gain is larger because the GEMV is cheaper, so the fixed rotation overhead is a larger share of per-layer latency; sequential decoding then accumulates this over long generations. Results on more GPUs are provided in Appendix G.3.

Table 4: Comparison of resource usage across ParoQuant, ReasoningQAT, and LAQuant.

			ParoQuant			ReasoningQAT			LAQuant		
Model	Batch	Shard.	Train tokens	VRAM (GB)	Time (h)	Train tokens	VRAM (GB)	Time (h)	Train tokens	VRAM (GB)	Time (h)
Q3-1.7B	32		167.8M	113.9	3.29	285.2M	79.79	30.96	167.8M	102.2	1.82
Q3-4B	16		167.8M	138.7	8.76	285.2M	98.03	66.41	167.8M	126.5	4.63
Q3-8B	16	✓	167.8M	121.3	19.82	285.2M	126.3	78.44	167.8M	119.2	14.91

Training. Table 4 compares training efficiency across ParoQuant, ReasoningQAT, and LAQuant on a single H200 144GB GPU. For a fair comparison with ParoQuant at matched accuracy, both methods use the same data budget (4,096 samples, length 2,048, 20 epochs); ReasoningQAT already uses 1.7× more tokens under its original two-stage training pipeline, so we keep both stages in its prescribed configuration. Each method runs at its largest feasible batch size, with *Shard.* denoting CPU offloading. LAQuant trains faster across all model sizes thanks to its lightweight optimization (no online-transform parameters, fewer tokens than ReasoningQAT); on Qwen3-8B, training drops from 19.82 to 14.91 GPU hours (24.8% reduction) at slightly lower peak VRAM.

5.3 Ablation Study

We conduct ablation studies under W3G128 quantization. Unless otherwise specified, we use 4,096 training samples, a maximum sequence length of 2,048, one-layer lookahead ($k = 1$), and 20 training epochs per layer, varying only the target factor in each study. Additional ablations are provided in Appendix I.

Table 5: Method ablations on Qwen3-1.7B (W3G128); (b) also evaluated on Qwen3-4B.

(a) Lookahead depth k: Pass@1 on AIME25 / LSAT, with their average.

	k=0	k=0→1	k=1	k=2
AIME25	26.88	28.39	29.33	28.44
LSAT	52.15	52.36	54.86	53.94
Avg.	39.52	40.38	**42.10**	41.19

(b) Per-projection output RMSE on AIME25 traces, with (k=1) vs without (k=0) lookahead.

Model	k	**Q**	**K**	**V**	**O**	**Gate**	**Up**	**Down**
1.7B	0	1.8729	0.4935	1.5039	0.8288	0.3675	0.3678	1.2166
	1	1.8442	0.4892	1.4812	0.8089	0.3590	0.3615	1.2025
	Δ	−1.53%	−0.88%	−1.51%	−2.40%	−2.31%	−1.70%	−1.16%
4B	0	0.2821	0.3188	0.2545	0.1344	0.1919	0.1756	0.2502
	1	0.2783	0.3162	0.2490	0.1304	0.1893	0.1724	0.2411
	Δ	−1.34%	−0.81%	−1.79%	−2.95%	−1.41%	−1.81%	−3.64%

Lookahead Depth. Table 5a reports lookahead depth k (Section 4.4), with k=0 the pure layer-wise objective, $k\geq1$ a k-layer lookahead, and k=0→1 a staged schedule. Lifting k from 0 to 1 improves the AIME25 / LSAT average from 39.52 to 42.10; the staged schedule reaches only 40.38, and k=2 adds training cost without accuracy gain. We default to $k = 1$.

Per-Projection Output RMSE. With the lookahead loss, all seven projections improve by 0.8–3.6% (Table 5b). The largest reductions land on the attention and MLP output projections (o_proj −2.95%, down_proj −3.64% at Qwen3-4B), which contribute directly to the next-layer input and so receive the strongest reweighting under cross-layer co-adaptation (Section 4.4).

Effect of Direct K/V-Cache Supervision. Table 6a evaluates per-layer K-only, V-only, and joint K/V MSE losses added to the controlled JSD-based E2E objective. K-only supervision modestly improves accuracy over plain JSD, whereas V-only and joint supervision do not; all three variants remain substantially below LAQuant. Table 6b shows that joint K/V supervision reduces both key and

Table 6: Effect of direct K/V-cache supervision on Qwen3-1.7B under W3G128 quantization. The controlled E2E baseline uses a teacher–student JSD loss. Auxiliary K/V activation losses use MSE, with a JSD:MSE coefficient ratio of 1:0.02 to approximately balance their initial scales.

(a) Pass@1 on AIME25, LSAT-AR, and LiveCodeBench for the JSD-based E2E baseline with K-only, V-only, or joint K/V matching.

Method	AIME25	LSAT-AR	LCB	Avg.
E2E	6.62	26.14	4.48	12.41
E2E + K-cache MSE	8.60	27.69	5.72	14.00
E2E + V-cache MSE	6.09	26.63	4.27	12.33
E2E + KV-cache MSE	4.95	27.55	3.60	12.03
LAQuant	29.33	54.86	21.02	**35.07**

(b) K/V-cache reconstruction RMSE and cosine similarity for the JSD-based E2E baseline, E2E with joint K/V MSE, and LAQuant. Lower RMSE and higher cosine similarity indicate better cache preservation.

Method	Key RMSE	Value RMSE	Key Cos	Value Cos
LAQuant	0.4892	1.4812	0.98986	0.95778
E2E+KV MSE	0.6424	2.0975	0.97600	0.88527
E2E	0.7811	2.8554	0.96639	0.84826

value RMSE and increases their cosine similarity relative to plain JSD. LAQuant, however, achieves lower RMSE and higher cosine similarity for both caches without explicit K/V supervision. The reduction in cache error under joint supervision is not accompanied by an improvement in reasoning accuracy. This result is consistent with our anisotropy analysis: minimizing aggregate activation error does not necessarily preserve the directions most relevant to downstream predictions. Together, these findings indicate that direct cache matching within an E2E objective does not replicate the effect of LAQuant's local lookahead supervision.

Table 7: Ablation studies on LAQuant design choices, including training set size and training sequence length.

(a) Training set size.

	1024	2048	4096	8192
AIME25	24.17	27.03	29.33	25.89
LSAT	47.88	52.26	54.86	54.10
Avg.	36.03	39.65	**42.10**	40.00

(b) Training sequence length.

	1024	2048	4096	8192
AIME25	29.12	29.33	29.53	27.24
LSAT	52.42	54.86	51.68	50.84
Avg.	40.77	**42.10**	40.61	39.04

Training Set Size. As shown in Table 7a, increasing the number of training samples consistently improves the average accuracy from 36.03 to 42.10 when the training set size is increased from 1,024 to 4,096. However, further increasing the size to 8,192 lowers the average accuracy to 40.00, indicating that the gain saturates beyond 4,096 samples. Therefore, we use 4,096 training samples as the default setting.

Training Sequence Length. As shown in Table 7b, increasing the sequence length from 1,024 to 2,048 improves the average accuracy. However, the gain saturates beyond 2,048 tokens, and further increasing the sequence length to 4,096 reduces the average accuracy to 40.61. Since longer sequences also increase training cost, 2,048 provides the best trade-off and is used as the default maximum sequence length.

6 Conclusion

We presented LAQuant, a single-stage layer-wise weight-only QAT method for large reasoning models. Our gradient-direction analysis identifies two factors driving the LRM-quantization difficulty gap: end-to-end supervision attenuates per-layer learning signals (including for KV-cache directions) via the softmax Fisher metric, and calibration-data choice is governed by Hessian-subspace alignment with the deployment distribution. LAQuant addresses both with a one-layer lookahead loss whose implicit cross-layer co-adaptation preserves the next-layer residual stream, paired with a reasoning-domain calibration corpus, producing standard integer weights deployable on any existing weight-only kernel. On Qwen3 and DeepSeek-R1-distilled Llama models, LAQuant attains the highest average reasoning accuracy among weight-only methods at W3G128 and W4G128 while delivering a 3.42$\times$ decoding speedup over FP16 on RTX A6000.

Acknowledgments and Disclosure of Funding

This work was supported by Samsung Electronics Co., Ltd. (IO241219-11552-01).

References

[1] Krish Agarwal, Rishi Astra, Adnan Hoque, Mudhakar Srivatsa, Raghu Ganti, Less Wright, and Sijia Chen. Hadacore: Tensor core accelerated hadamard transform kernel. *arXiv preprint arXiv:2412.08832*, 2024.

[2] 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. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

[3] Hong Chen, Chengtao Lv, Liang Ding, Haotong Qin, Xiabin Zhou, Yifu Ding, Xuebo Liu, Min Zhang, Jinyang Guo, Xianglong Liu, et al. Db-llm: Accurate dual-binarization for efficient llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8719–8730, 2024.

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

[5] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 2924–2936, 2019.

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

[7] Tri Dao. Fast Hadamard Transform in CUDA, with a PyTorch Interface. `https://github.com/Dao-AILab/fast-hadamard-transform`, 2024.

[8] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[9] Xin Ding, Xiaoyu Liu, Zhijun Tu, Yun Zhang, Wei Li, Jie Hu, Hanting Chen, Yehui Tang, Zhiwei Xiong, Baoqun Yin, and Yunhe Wang. CBQ: Cross-block quantization for large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

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

[11] Razvan-Gabriel Dumitru, Vikas Yadav, Rishabh Maheshwary, Paul-Ioan Clotan, Sathwik Tejaswi Madhusudhan, and Mihai Surdeanu. Layer-wise quantization: A pragmatic and effective method for quantizing llms beyond integer bit-levels. *arXiv preprint arXiv:2406.17415*, 2024.

[12] Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 35:4475–4488, 2022.

[13] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[14] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024.

[15] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[16] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[17] Etash Kumar Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Rea Sprague, et al. OpenThoughts: Data recipes for reasoning models. In *International Conference on Learning Representations (ICLR)*, 2026.

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

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

[20] Babak Hassibi and David Stork. Second order derivatives for network pruning: Optimal brain surgeon. *Advances in neural information processing systems*, 5, 1992.

[21] Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, and Xiaojuan Qi. Billm: pushing the limit of post-training quantization for llms. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[22] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

[23] Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025.

[24] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

[25] Maxwell Jia. Aime problem set 2024, 2024.

[26] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. Squeezellm: Dense-and-sparse quantization. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[27] Medha Kumar, Zifei Xu, Xin Wang, and Tristan Webb. The impact of quantization on large reasoning model reinforcement learning. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025.

[28] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[29] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

[30] Geonho Lee, Janghwan Lee, Sukjin Hong, Minsoo Kim, Euijai Ahn, Du-Seong Chang, and Jungwook Choi. Rilq: Rank-insensitive lora-based quantization error compensation for boosting 2-bit large language model accuracy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18091–18100, 2025.

[31] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations (ICLR)*, 2021.

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

[33] Yesheng Liang, Haisheng Chen, Zihan Zhang, Song Han, and Zhijian Liu. ParoQuant: Pairwise Rotation Quantization for Efficient Reasoning LLM Inference. In *International Conference on Learning Representations (ICLR)*, 2026.

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

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

[36] Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *Proceedings of Machine Learning and Systems*, 7, 2025.

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

[38] Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. Vptq: Extreme low-bit vector post-training quantization for large language models. In *The 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

[39] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: LLM quantization with learned rotations. In *The Thirteenth International Conference on Learning Representations*, 2025.

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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

[42] math ai. Aime problem set 2025, 2025.

[43] Anmol Mekala, Anirudh Atmakuru, Yixiao Song, Marzena Karpinska, and Mohit Iyyer. Does quantization affect models' performance on long-context tasks? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9433–9481, 2025.

[44] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

[45] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332, 2025.

[46] Yasuyuki Okoshi, Hikari Otsuka, Daichi Fujiki, and Masato Motomura. Towards quantization-aware training for ultra-low-bit reasoning LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

[47] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[48] 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 *First Conference on Language Modeling*, 2024.

[49] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[51] Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[52] Albert Tseng, Qingyao Sun, David Hou, and Christopher De. Qtip: Quantization with trellises and incoherence processing. *Advances in Neural Information Processing Systems*, 37:59597–59620, 2024.

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

[54] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[55] Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. Redpajama: an open dataset for training large language models. *NeurIPS Datasets and Benchmarks Track*, 2024.

[56] Miles Williams and Nikolaos Aletras. On the impact of calibration data in post-training quantization and pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10100–10118, 2024.

[57] Yang Xiang, Yixin Ji, Juntao Li, and Min Zhang. Think before you prune: Selective self-generated calibration for pruning large reasoning models, 2025.

[58] 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 *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[59] 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.

[60] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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

[62] 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.

[63] 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 P. Gibbons, G. Pekhimenko, and C. De Sa, editors, *Proceedings of Machine Learning and Systems*, volume 6, pages 196–209, 2024.

[64] Xingyu Zheng, Yuye Li, Haoran Chu, Yue Feng, Xudong Ma, Zining Wang, Jie Luo, Jinyang Guo, Haotong Qin, Michele Magno, and Xianglong Liu. An empirical study of qwen3 quantization. *Visual Intelligence*, 4(1):11, 2026.

[65] Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. Ar-lsat: Investigating analytical reasoning of text. *arXiv preprint arXiv:2104.06598*, 2021.

A Related Work

Post-training quantization (PTQ) for LLMs. GPTQ [13] introduced layer-wise OBS-style weight reconstruction with per-group quantization, becoming the dominant PTQ recipe for ≥ 4-bit weight-only quantization. AWQ [35] adds per-channel scaling absorbed into the preceding LayerNorm to mitigate activation outliers. SqueezeLLM [26] introduces sensitivity-based non-uniform quantization paired with a dense-and-sparse decomposition for outlier handling. SmoothQuant [58] and OmniQuant [49] extend the channel-scaling and clipping ideas with learnable parameters. Rotation-based methods including QuaRot [2], SpinQuant [39], and DuQuant [34] insert orthogonal transforms (typically Hadamard) to mix outlier directions into the bulk before quantization, achieving 4-bit all-quantization but introducing online rotation overhead at inference. Variable-bit-per-layer schemes [11] assign different bit widths across layers based on layer-importance signals, complementing fixed-bit recipes. Lower-bit weight-only methods reach 2-bit [4, 10, 3, 38] or even 1-bit [21, 53] regimes.

Quantization-aware training (QAT) for LLMs. EfficientQAT [4] introduced block-wise QAT with learnable quantization parameters followed by an optional end-to-end (E2E) phase, demonstrating that scales and zeros can be trained jointly with weights at scale. BitDistiller [10] and RILQ [30] apply teacher–student distillation in an E2E fashion. SpinQuant [39] performs an additional E2E phase over rotation parameters.

Quantization for large reasoning models. ReasoningQAT [46] chains layer-wise and E2E QAT with a domain-mixed calibration corpus, the first quantization pipeline targeting LRMs specifically. ParoQuant [33] combines minimal pairwise rotation with channel scaling to reduce per-layer error. QuantLRM [62] leverages weight-update statistics from reasoning-incentivized fine-tuning to identify per-channel importance for LRM quantization. Concurrent observations [32, 37, 64, 43] report that legacy quantization recipes degrade markedly on long-decoding reasoning benchmarks; recent analyses [27] further study the interaction between quantization and reasoning-incentivized RL training.

Calibration data selection. Pre-training corpora (WikiText2 [44], C4 [47], RedPajama [55]) have been the default calibration source. Recent work [56] reports that domain-aligned calibration improves task accuracy, with effects largest at low bit widths and on instruction-tuned models. We extend this empirical observation with a Hessian-subspace-alignment formalization in Section 4.1.

B Proofs of Propositions 1–3 and Spectral Bound

We restate and prove the spectral-norm bound on H_{KL} used in Section 4.3.

Lemma 1 (Spectral-norm bound on the categorical Fisher) *For any $p^\star \in \Delta^{V-1}$,*

$$\sigma_{\max}\big(\mathrm{diag}(p^\star) - p^\star (p^\star)^\top\big) \;\leq\; 2 \max_i p_i^\star (1 - p_i^\star) \;\leq\; \tfrac{1}{2}, \tag{9}$$

with the upper bound vanishing as $\max_i p_i^\star \to 1$.

Proof 1 *Let $A = \mathrm{diag}(p^\star) - p^\star (p^\star)^\top$. The diagonal entries are $A_{ii} = p_i^\star(1 - p_i^\star)$ and the off-diagonals are $A_{ij} = -p_i^\star p_j^\star$. Since $A \succeq 0$, $\sigma_{\max}(A)$ equals its largest eigenvalue. By Gershgorin's circle theorem,*

$$\sigma_{\max}(A) \;\leq\; \max_i \Big(A_{ii} + \sum_{j \neq i} |A_{ij}| \Big) \;=\; \max_i \Big(p_i^\star(1 - p_i^\star) + p_i^\star \sum_{j \neq i} p_j^\star \Big) \;=\; 2 \max_i p_i^\star (1 - p_i^\star),$$

using $\sum_{j \neq i} p_j^\star = 1 - p_i^\star$. The map $t \mapsto t(1-t)$ on $[0,1]$ attains its maximum $1/4$ at $t = 1/2$, giving $2 \max_i p_i^\star(1 - p_i^\star) \leq 1/2$. As $\max_i p_i^\star \to 1$, all probabilities approach 0 or 1, so $\max_i p_i^\star(1 - p_i^\star) \to 0$ and the bound vanishes.

Proposition 3 (E2E gradient attenuation) *Combining Proposition 1 with Lemma 1, every direction v in the per-layer residual space satisfies*

$$\|M_{\mathrm{E2E}}\, v\| \;\leq\; 2 \max_i p_i^\star (1 - p_i^\star) \;\cdot\; \sigma_{\max}(\mathcal{J})^2 \;\cdot\; \|v\|, \tag{10}$$

whereas $\|M_{\mathrm{lw}}\, v\| = \|v\|$. Inequality (10) *bounds the operator norm of M_{E2E} by the product of the categorical-Fisher spectral norm and the squared spectral norm of the downstream Jacobian: when $p^\star$ is sharply peaked and $\mathcal{J}$ is non-expansive, both factors are small and E2E gradients can be strongly attenuated, with the second factor decaying geometrically only when the downstream Jacobians are strictly contractive on average across depth.*

Proof 2 (Proof of Proposition 1) *The layer-wise reconstruction loss $\mathcal{L}_{\mathrm{lw}} = \frac{1}{2}\|\varepsilon_\ell\|^2$ has exact Hessian I in ε_ℓ, so $M_{\mathrm{lw}} = I$. For E2E, the forward-KL loss against the teacher's softmax distribution $p^\star$ (equivalently, cross-entropy with the fixed target $p^\star$) has Hessian $H_{\mathrm{KL}} = \mathrm{diag}(p^\star) - p^\star p^{\star\top}$ in the pre-softmax logits z (the categorical Fisher information of $p^\star$). Composing with $\mathcal{J} = \partial z/\partial \varepsilon_\ell$ via the chain rule, the Hessian of the E2E loss in ε_ℓ is $\mathcal{J}^\top H_{\mathrm{KL}} \mathcal{J} + \sum_i (p_i - p_i^\star)\, \nabla^2_{\varepsilon_\ell} z_i$. The local Gauss–Newton approximation drops the residual second-derivative term — which vanishes exactly when the student matches the teacher downstream and is small in a neighborhood of that point — giving $M_{\mathrm{E2E}} = \mathcal{J}^\top H_{\mathrm{KL}} \mathcal{J}$.*

Proof 3 (Proof of Proposition 2) *Taylor-expanding $F_{\ell+1}$ around $F_\ell(h_\ell^\star)$ gives $F_{\ell+1}(F_\ell(h_\ell^\star)+\varepsilon_\ell) - F_{\ell+1}(F_\ell(h_\ell^\star)) = J_{\ell+1}\,\varepsilon_\ell + O(\|\varepsilon_\ell\|^2)$, where $\varepsilon_\ell = Q_\ell(\hat{h}_\ell; \theta_q^{(\ell)}) - F_\ell(h_\ell^\star)$ and $J_{\ell+1} = \partial F_{\ell+1}/\partial h$ at $h = F_\ell(h_\ell^\star)$. Substituting into $\mathcal{L}_{\mathrm{la}}$, which reduces to $\frac{1}{2}\|\cdot\|^2$ below the Huber threshold (Section 4.4), yields the local quadratic form $\frac{1}{2}\,\varepsilon_\ell^\top\, J_{\ell+1}^\top J_{\ell+1}\,\varepsilon_\ell + O(\|\varepsilon_\ell\|^3)$, hence $M_{\mathrm{la}} = J_{\ell+1}^\top J_{\ell+1}$.*

Proof 4 (Proof of Proposition 3) *By Proposition 1 and sub-multiplicativity of operator norms, $\|M_{\mathrm{E2E}}\, v\| \le \sigma_{\max}(\mathcal{J})^2\, \sigma_{\max}(H_{\mathrm{KL}})\, \|v\|$. Applying Lemma 1 to bound $\sigma_{\max}(H_{\mathrm{KL}})$ gives the stated inequality. The LW case is immediate from $M_{\mathrm{lw}} = I$.*

C Direction-Stratified Anisotropy and Top-1 Agreement

We empirically distinguish two scenarios consistent with the operator-norm bound of Section 4.3: *(i) uniform shrinkage* (compensable by a larger learning rate), or *(ii) anisotropic attenuation* where the leading subspace of M_{E2E} is small and the rest — including KV-cache-reducing directions — decays sharply outside it (no scalar reweighting can repair it). Direct measurement on AIME25 traces confirms (ii). A token-weighted top-1 agreement comparison against the BF16 teacher provides the corresponding downstream confirmation.

Setup. We compare the LW (k=0) student of Table 5a and a controlled-E2E baseline (Qwen3-1.7B W3G128, trained from the same BF16 layer-wise checkpoint and calibration corpus as LAQuant but with E2E loss in place of the layer-wise objective) on Qwen3-1.7B; the only thing differing between the two is the supervision. For each of 16 AIME25 responses $\times$ 6 highest-entropy positions $\times\ \ell \in \{4, 8, 12, 16, 20, 24\}$ (453 valid triples after OOMs at small ℓ) we measure $\|M_{\mathrm{E2E}} v\|/\|v\|$ at $v \in \{v_\star, v_r, g_K^{\mathrm{LW}}, g_V^{\mathrm{LW}}, g_K^{\text{ctrl-E2E}}, g_V^{\text{ctrl-E2E}}\}$, where $v_\star$ is the leading eigenvector of M_{E2E} from ≤ 20 power-iter steps, v_r a random Gaussian, and $g_{K/V}^X = \nabla_{h_\ell[p]} \|K_{\ell+1}^X(h_\ell) - K_{\ell+1}^*(h_\ell)\|^2$ is the residual direction that maximally reduces next-layer K- (resp. V-) cache error for student X. JVP via `torch.func.jvp` (exact, eager attention); VJP via reverse-mode AD; causal sequence truncation to $[:p+1]$. Entropy stratification ($H \ge 2$ nats) avoids the near-deterministic regime where H_{KL} is null-rank in finite precision.

Anisotropy of M_{E2E}. Table 8 reports the medians: the leading eigendirection of M_{E2E} is $\sim$30–40$\times$ more responsive than every other direction we test, and the K/V-cache-reducing directions sit in the attenuated regime alongside random Gaussians. By contrast, $M_{\mathrm{LW}} = I$ (Proposition 1) assigns responsiveness exactly 1.0 to every direction. The observed $\ge 30\times$ ratio rules out the uniform-shrinkage alternative, which would require every direction's responsiveness to be near 1.0 relative to $v_\star$ (and would in principle be compensable by a learning-rate scaling); we observe ≤ 0.05 at the median for every non-leading direction tested, with the pattern robust across all six ℓ values.

$\sigma_{\max}(\mathcal{J})$ and bound looseness. Direct power iteration on $\mathcal{J}^\top \mathcal{J}$ gives median $\sigma_{\max}(\mathcal{J}) = 40.8$ (range 9.1–479.2, growing with the count of downstream layers): $\mathcal{J}$ is mildly expansive in this regime. The operator-norm bound's RHS $2 \max_i p_i^\star(1 - p_i^\star) \cdot \sigma_{\max}(\mathcal{J})^2$ therefore exceeds the directly measured $\lambda_{\max}(M_{\mathrm{E2E}})$ by 4–5 orders of magnitude — the bound holds but is loose, since it

assumes worst-case alignment between $\mathcal{J}$'s leading singular vector and H_{KL}'s leading eigenvector that holds with probability $O(1/V)$ in generic alignment. The dilution claim does not depend on the bound's tightness: the direction-stratified ratio is invariant to the absolute scale of $\sigma_{\max}(\mathcal{J})$, which cancels as a common multiplier.

Cross-student magnitudes. The controlled-E2E student's residual K-cache gradient is $1.52\times$ larger than the LW student's (median $\|g_K^{\text{ctrl-E2E}}\|/\|g_K^{\text{LW}}\|$, IQR 1.40–1.69); the V-cache ratio is 1.33 (IQR 1.22–1.48). E2E supervision left ∼50% more residual K-cache error than LW under matched training, on a residual direction M_{E2E} attenuates ∼30× regardless of student ($\cos(g_K^{\text{LW}}, g_K^{\text{ctrl-E2E}}) = 0.67$, IQR 0.46–0.74). $M_{\text{LW}} = I$ avoids this failure mode entirely.

Top-1 agreement with the BF16 teacher. Table 9 reports the fraction of next-token positions where the BF16 teacher's argmax matches each model's argmax, weighted by tokens. LAQuant attains the highest agreement at 94.14%, followed by LW ($k = 0$, LAQuant without the lookahead loss) at 93.95%, the original E2E baseline (ReasoningQAT) at 93.07%, and the controlled E2E baseline at 88.44%. The controlled-E2E gap (which isolates the loss-shape variable from calibration data and initialization, both shared with LAQuant) supports the dilution argument: holding everything else fixed, switching from LW to E2E supervision *reduces* fidelity to the BF16 teacher's predictions, despite the E2E loss explicitly optimizing for that fidelity. The LA−LW gap is small in aggregate ($+0.20$ pp) but concentrates at high-entropy positions where decisions actually compete (Table 10): negligible at $H < 0.1$ ($+0.004$ to $+0.050$ pp, ∼56% of tokens) but reaching $+0.77$ pp at $H \in [1, 2)$ and $+2.50$ pp at $H \in [2, 5)$.

Table 8: Median (IQR) of $\|M_{\text{E2E}}v\|/\|M_{\text{E2E}}v_\star\|$ across 453 (sample, position, layer) triples on Qwen3-1.7B AIME25 traces. By Proposition 1, $M_{\text{LW}} = I$ assigns responsiveness 1.0 to every direction.

direction	median (q25, q75)
$v_\star$ (leading eigvec)	1.000
random v_r	0.025 (0.017, 0.034)
g_K^{LW}	0.037 (0.025, 0.053)
$g_K^{\text{ctrl-E2E}}$	0.037 (0.026, 0.055)
g_V^{LW}	0.027 (0.019, 0.037)
$g_V^{\text{ctrl-E2E}}$	0.029 (0.020, 0.042)

Table 9: Token-weighted top-1 argmax agreement with the BF16 teacher on Qwen3-1.7B AIME25 traces (488,455 tokens, 120 traces).

Model (W3G128)	Agreement
LAQuant (k=1)	**94.14%**
LW (k=0)	93.95%
E2E (ReasoningQAT)	93.07%
E2E-Ctrl	88.44%

Table 10: Top-1 agreement vs the BF16 teacher, stratified by teacher entropy H. The LA−LW gap grows with H: at sharply peaked positions ($H < 0.1$, ∼56% of tokens) LA and LW are indistinguishable; at uncertain positions ($H \geq 1$) LA preserves the teacher's choice noticeably more often.

H range	frac.	LA	LW	E2E	LA−LW
$[0, 0.01)$	45.6%	99.98%	99.97%	99.94%	+0.004
$[0.01, 0.1)$	10.6%	99.81%	99.76%	99.57%	+0.050
$[0.1, 0.5)$	15.1%	98.95%	98.75%	98.02%	+0.199
$[0.5, 1.0)$	16.6%	86.29%	85.92%	83.83%	+0.372
$[1.0, 2.0)$	11.6%	72.97%	72.20%	68.94%	**+0.766**
$[2.0, 5.0)$	0.6%	58.21%	55.70%	50.40%	**+2.502**

D Peaky Softmax Distribution on AIME25

We empirically validate that the BF16 teacher's softmax distribution on LRM decoding is sharply peaked, so the spectral bound $\sigma_{\max}(H_{\mathrm{KL}}) \leq 2\max_i p_i^\star(1 - p_i^\star)$ used in Section 4.3 is small in practice.

We collect 488,455 next-token softmax distributions from Qwen3-1.7B (BF16) decoding 120 AIME25 traces. The teacher's top-1 probability has mean 0.877 and median 0.996; 67.6% of tokens have $\max_i p_i^\star > 0.9$ and 53.4% have $\max_i p_i^\star > 0.99$. By Lemma 1, for the latter group $\sigma_{\max}(H_{\mathrm{KL}}) \leq$

$2 \cdot 0.99 \cdot 0.01 \approx 0.02$, more than an order of magnitude below the worst-case bound of $1/2$ (a factor of $\sim 25\times$).

E Top-k Sensitivity of Hessian-Subspace Overlap

The principal-angle diagnostic of Section 4.1 uses the average top-k cosine $\bar{c}_k$ between $H_{\text{act}}^{\text{cal}}$ and $H_{\text{act}}^{\text{eval}}$, with $k = 64$ in Figure 3b(b) ($\sim$3% of the d=2048 eigenspace). We sweep $k \in \{4, 8, 16, 32, 64, 128, 256\}$ to confirm the calibration ranking does not depend on this choice.

Following the Figure 3b setup, we compute $\bar{c}_k$ across 112 (layer, hook) keys (28 transformer layers $\times$ 4 attention/MLP hook points) on Qwen3-1.7B activations, between AIME25 and four candidate calibration corpora. Table 11 reports the mean cosine overlap.

Setup. For each of 4 hook sites per transformer layer on Qwen3-1.7B (28 layers $\times$ 4 sites = 112 keys; the four sites are the inputs to `q_proj` (attention input, shared by q/k/v), `o_proj` (attention output), `gate_proj` (MLP input, shared by gate/up), and `down_proj` (MLP hidden)), we collect activations from 128 sequences $\times$ 2,048 tokens per corpus and accumulate the uncentered second-moment matrix $H_{\text{act}} = \sum_t \phi_t \phi_t^\top$. Eigendecomposition uses `torch.linalg.eigh` truncated to the top 256 eigenvectors per (corpus, key); per-pair cosines are arithmetic-meaned across the 112 keys. Table 11 reports the resulting mean cosine overlap between AIME25 and four candidate calibration corpora.

The ranking OpenR1-Math $>$ OpenThoughts3 $>$ ReasoningQAT $>$ WikiText2+C4+RedPajama holds at every $k \in [4, 256]$, and the reasoning-vs-generic gap stays ≥ 0.21 cosine units throughout. A mild U-shape minimized near $k \approx$ 32–64 reflects that small k over-weights a few high-variance directions, while large k admits low-variance directions that are nearly uniformly shared across corpora; $k = 64$ sits at the discriminating regime.

Table 11: Mean top-k principal-angle cosine $\bar{c}_k$ between four calibration corpora and the AIME25 deployment distribution on Qwen3-1.7B, averaged across 112 (layer, hook) keys. W2 and RP denote WikiText2 and RedPajama, respectively.

Calibration corpus	k=4	k=8	k=16	k=32	k=64	k=128	k=256
OpenR1-Math (LAQuant)	0.947	0.915	0.886	0.864	**0.846**	0.836	0.835
OpenThoughts3	0.757	0.721	0.701	0.691	**0.686**	0.695	0.716
ReasoningQAT mix	0.711	0.689	0.674	0.669	**0.666**	0.676	0.697
W2+C4+RP (generic)	0.476	0.468	0.450	0.451	**0.455**	0.478	0.517

F Details of Evaluation Settings

Evaluation Settings. We use lighteval [19] for reasoning and lm_eval [14] for non-reasoning benchmarks, both on vLLM v0.10.1.1 [28]. We follow each model's original sampling: temperature 0.6 with `top_p`=0.95, plus `top_k`=20 for Qwen3.

For AIME24, AIME25, GPQA, and LSAT-AR, we generate 16 responses for each problem and compute Pass@1 from the generated samples. For MMLU-Pro and LiveCodeBench, we generate a single response per problem and compute Pass@1 accordingly. Due to the limited number of problems in AIME24 and AIME25, which leads to relatively high statistical variability, we report results averaged over four random seeds, 0, 1, 2, and 3, to obtain more reliable estimates. For LiveCodeBench, we also report the average over three random seeds, 0, 1, and 42. For all other reasoning tasks, we use a fixed random seed of 42.

For DeepSeek-R1-distilled Llama-3.1, a known issue has been reported in which the model may skip the thinking process even when evaluated in thinking mode. To avoid this issue, we explicitly prepend the `<think>` token during all evaluations[4].

Contamination Check of QAT Data. To rule out training–evaluation contamination, we check whether any AIME24 or AIME25 problem used in our math-reasoning evaluation appears among the

[4] `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B#usage-recommendations`

4,096 OpenR1-Math-220k [23] samples used for QAT. Table 12a breaks these samples down by the `source` field of OpenR1-Math-220k: most come from `olympiads` (72.0%), and only 164 (4.0%) come from `amc_aime`. Table 12b tallies the 33 AIME problems among the `amc_aime` samples by competition year. All of them date from 1984–2022 and none is from 2023 or later, confirming the disjointness from AIME24 and AIME25 stated in the implementation details (Section 5).

Table 12: Composition of the 4,096 QAT samples drawn from OpenR1-Math-220k.

(a) Samples by `source` field.

Source	Count
`olympiads`	2,951
`cn_contest`	545
`aops_forum`	367
`amc_aime`	164
`olympiads_ref`	35
`inequalities`	17
`number_theory`	17
Total	4,096

(b) AIME problems among the `amc_aime` samples, by competition year.

Year	Count	Year	Count	Year	Count
1984	2	2001	1	2012	3
1986	1	2002	1	2015	1
1988	1	2004	1	2016	1
1991	2	2007	1	2018	1
1992	1	2009	2	2019	2
1998	1	2010	3	2020	1
1999	1	2011	1	2022	1
2000	4				
Total: 33 (none from 2023 or later)					

G Additional Evaluation Results

G.1 Reasoning Tasks

Sample budget and variability. For AIME24, AIME25, GPQA, and LSAT-AR, we generate 16 responses per problem to estimate Pass@1. Since AIME24 and AIME25 each contain only 30 competition problems, their estimates exhibit relatively high run-to-run variability. We therefore additionally average the results over four random seeds $(0, 1, 2, 3)$, yielding a total of 64 samples per problem for these benchmarks. Table 13 reports the mean and standard deviation of Pass@1 for each method, allowing readers to assess the statistical variability together with the central estimate.

ReasoningQAT (R-QAT) training data. R-QAT uses a public configuration whose calibration data differ from those of LAQuant. In phase 1, corresponding to layer-wise QAT, R-QAT uses a mixture of reasoning data (OpenThoughts-3) and generic calibration data with 4,096 samples and a maximum sequence length of 2,048. In phase 2, corresponding to end-to-end QAT, it further uses reasoning-only data from OpenThoughts-3 with 32,768 samples and a maximum sequence length of 8,192. Thus, although the calibration corpus is not exactly matched to ours, R-QAT is trained with a substantially larger number of reasoning-domain training tokens than LAQuant. Because reproducing both phases under a fully controlled calibration setup is non-trivial, we evaluate R-QAT using its prescribed public configuration in all comparisons. We instead provide the matched-calibration comparison through ParoQuant++, which uses the same reasoning-domain corpus and training data budget as LAQuant.

Table 13: Pass@N$^{\uparrow}$ and standard deviations on **AIME**, **GPQA Diamond**, and **LSAT-AR** under 3-bit and 4-bit quantization. R-QAT and ParoQ denote ReasoningQAT and ParoQuant, respectively. Results on AIME benchmarks are averaged over four random seeds, 0, 1, 2, and 3.

(a) W3G128 Pass@N accuracy on reasoning tasks.

Model	Method	Dataset	AIME24		AIME25		GPQA		LSAT	
			Pass@1	Pass@64	Pass@1	Pass@64	Pass@1	Pass@16	Pass@1	Pass@16
Qwen3-1.7B	FP16		45.84 ± 6.20	83.30	35.94 ± 4.58	73.30	40.70 ± 2.08	78.28	64.10 ± 2.53	95.65
	GPTQ	Generic	0.00 ± 0.00	0.00	0.00 ± 0.00	0.00	25.03 ± 3.05	81.82	18.78 ± 2.72	76.96
	GPTQ++	OpenR1	17.14 ± 4.48	60.00	22.55 ± 4.35	46.70	28.88 ± 1.77	72.22	34.40 ± 2.36	80.43
	R-QAT	Mixed	25.52 ± 6.26	73.30	23.70 ± 4.79	60.00	31.03 ± 2.68	75.76	47.69 ± 2.09	95.22
	ParoQ	Generic	7.24 ± 4.77	53.30	16.88 ± 4.40	46.70	30.49 ± 3.19	76.26	36.10 ± 2.68	81.30
	ParoQ++	OpenR1	33.65 ± 5.93	73.30	28.23 ± 4.28	56.70	30.81 ± 2.61	76.26	49.30 ± 2.38	93.48
	LAQuant	OpenR1	34.64 ± 6.10	80.00	29.33 ± 4.87	60.00	31.79 ± 2.16	75.76	54.86 ± 2.11	93.91
Qwen3-4B	FP16		73.23 ± 4.52	90.00	63.75 ± 5.65	86.67	55.59 ± 2.20	82.32	81.14 ± 1.21	95.22
	GPTQ	Generic	12.92 ± 4.46	36.67	12.76 ± 3.59	40.00	33.65 ± 2.56	74.75	40.41 ± 1.78	86.09
	GPTQ++	OpenR1	54.95 ± 5.73	76.67	51.21 ± 4.67	80.00	41.60 ± 2.02	76.77	67.45 ± 1.40	90.87
	R-QAT	Mixed	43.75 ± 6.04	83.33	35.31 ± 4.74	76.67	46.21 ± 2.91	78.79	67.64 ± 2.67	94.78
	ParoQ	Generic	41.30 ± 5.50	73.33	39.58 ± 5.60	80.00	44.41 ± 2.27	74.24	68.94 ± 1.86	90.00
	ParoQ++	OpenR1	59.74 ± 6.73	83.33	52.76 ± 4.70	83.33	46.17 ± 2.07	80.23	77.07 ± 1.85	94.35
	LAQuant	OpenR1	62.92 ± 5.39	90.00	54.69 ± 5.56	86.67	46.21 ± 2.11	77.78	77.15 ± 1.94	96.09
Qwen3-8B	FP16		74.95 ± 4.24	93.33	67.66 ± 5.81	86.67	58.78 ± 2.49	87.88	85.57 ± 1.24	95.65
	GPTQ	Generic	44.48 ± 6.46	73.33	40.68 ± 6.09	76.70	48.17 ± 1.69	79.80	69.92 ± 2.41	92.61
	GPTQ++	OpenR1	68.08 ± 4.96	86.67	59.43 ± 6.07	83.30	51.20 ± 1.80	82.83	82.28 ± 1.93	96.09
	R-QAT	Mixed	53.75 ± 5.83	83.33	43.91 ± 6.13	86.70	48.61 ± 2.08	83.84	77.23 ± 2.11	94.78
	ParoQ	Generic	59.53 ± 5.50	80.00	52.92 ± 6.13	83.30	52.30 ± 1.85	82.83	78.53 ± 1.34	95.65
	ParoQ++	OpenR1	67.19 ± 5.76	90.00	58.39 ± 6.44	83.30	53.09 ± 2.60	86.36	79.48 ± 1.87	96.96
	LAQuant	OpenR1	68.33 ± 4.75	93.33	61.30 ± 6.10	86.67	53.66 ± 2.02	85.86	83.80 ± 1.94	96.09
R1-Distill-Llama-8B	FP16		43.44 ± 6.45	80.00	31.20 ± 5.32	70.00	48.17 ± 2.17	86.87	59.78 ± 2.97	96.52
	GPTQ	Generic	15.45 ± 6.21	73.33	19.64 ± 4.03	60.00	37.97 ± 2.57	85.35	42.47 ± 3.09	95.22
	GPTQ++	OpenR1	25.84 ± 6.26	83.33	25.73 ± 5.02	66.67	37.78 ± 2.48	84.34	46.36 ± 2.73	93.48
	R-QAT	Mixed	30.16 ± 6.78	83.33	27.50 ± 4.44	66.67	40.50 ± 2.46	82.83	48.48 ± 2.32	94.78
	ParoQ	Generic	0.89 ± 1.81	13.33	0.05 ± 0.13	3.33	32.99 ± 2.66	88.38	29.65 ± 2.32	93.48
	ParoQ++	OpenR1	30.63 ± 6.79	76.67	25.57 ± 4.90	56.67	40.37 ± 2.38	86.36	48.89 ± 2.94	95.65
	LAQuant	OpenR1	31.46 ± 7.34	83.33	28.54 ± 3.80	70.00	43.31 ± 2.27	84.85	50.98 ± 2.43	92.61

(b) W4G128 Pass@N on reasoning tasks.

Model	Method	Dataset	AIME24		AIME25		GPQA		LSAT	
			Pass@1	Pass@64	Pass@1	Pass@64	Pass@1	Pass@16	Pass@1	Pass@16
Qwen3-1.7B	FP16		45.84 ± 6.20	83.33	35.94 ± 4.58	73.33	40.70 ± 2.08	78.28	64.10 ± 2.53	95.65
	GPTQ	Generic	24.17 ± 5.60	60.00	25.47 ± 4.13	53.33	33.14 ± 2.06	77.78	48.29 ± 2.42	86.52
	GPTQ++	OpenR1	38.96 ± 6.78	80.00	33.33 ± 4.28	66.67	34.60 ± 2.24	72.73	61.03 ± 2.36	91.74
	R-QAT	Mixed	32.61 ± 6.49	83.33	27.29 ± 5.10	66.67	35.70 ± 2.32	76.77	53.48 ± 1.74	93.04
	ParoQ	Generic	39.53 ± 5.82	80.00	32.55 ± 4.63	66.67	35.86 ± 2.89	73.23	58.30 ± 2.45	92.17
	ParoQ++	OpenR1	41.82 ± 5.94	80.00	32.71 ± 4.58	66.67	36.30 ± 3.09	72.73	60.52 ± 2.33	91.74
	LAQuant	OpenR1	42.35 ± 6.52	80.00	35.47 ± 4.54	73.33	36.43 ± 1.64	76.77	61.14 ± 2.23	91.74
Qwen3-4B	FP16		73.23 ± 4.52	90.00	63.75 ± 5.65	86.67	55.59 ± 2.20	82.32	81.14 ± 1.21	95.22
	GPTQ	Generic	63.44 ± 4.75	80.00	58.67 ± 5.40	86.67	51.61 ± 1.77	82.32	78.10 ± 2.18	92.61
	GPTQ++	OpenR1	68.33 ± 4.90	90.00	62.20 ± 5.19	90.00	51.86 ± 2.37	77.78	78.61 ± 1.25	93.91
	R-QAT	Mixed	64.64 ± 5.56	86.67	57.66 ± 5.56	80.00	52.15 ± 1.27	78.79	78.45 ± 1.89	93.91
	ParoQ	Generic	67.29 ± 5.63	86.67	60.78 ± 5.12	90.00	52.46 ± 2.16	79.29	79.43 ± 1.77	93.48
	ParoQ++	OpenR1	70.37 ± 4.83	90.00	61.72 ± 5.28	83.33	53.44 ± 2.42	83.84	80.87 ± 1.34	95.22
	LAQuant	OpenR1	70.42 ± 4.81	90.00	62.66 ± 5.79	86.67	53.60 ± 1.65	81.31	81.03 ± 2.34	94.78
Qwen3-8B	FP16		74.95 ± 4.24	93.33	67.66 ± 5.81	86.67	58.78 ± 2.49	87.88	85.57 ± 1.24	95.65
	GPTQ	Generic	71.56 ± 5.34	83.33	65.21 ± 4.86	83.33	57.89 ± 2.35	82.32	82.31 ± 1.42	96.09
	GPTQ++	OpenR1	72.76 ± 5.02	93.33	65.16 ± 5.07	86.67	57.83 ± 2.11	85.35	83.89 ± 1.17	96.52
	R-QAT	Mixed	68.44 ± 4.97	93.33	59.48 ± 6.46	83.33	58.49 ± 2.07	86.87	85.24 ± 1.25	96.96
	ParoQ	Generic	73.39 ± 4.70	93.33	64.52 ± 5.58	86.67	56.85 ± 2.26	84.34	84.43 ± 1.43	95.22
	ParoQ++	OpenR1	73.81 ± 5.17	90.00	64.27 ± 5.23	90.00	56.66 ± 2.38	83.84	83.99 ± 1.30	96.09
	LAQuant	OpenR1	74.17 ± 4.48	93.33	66.98 ± 4.78	86.67	57.23 ± 2.53	86.36	85.24 ± 1.77	96.52
R1-Distill-Llama-8B	FP16		43.44 ± 6.45	80.00	31.20 ± 5.32	70.00	48.17 ± 2.17	86.87	59.78 ± 2.97	96.52
	GPTQ	Generic	38.44 ± 6.95	80.00	29.85 ± 5.05	66.67	46.65 ± 2.38	87.88	56.66 ± 2.68	95.22
	GPTQ++	OpenR1	39.53 ± 5.34	83.33	29.74 ± 4.42	63.33	45.30 ± 2.42	84.85	58.40 ± 2.17	95.22
	R-QAT	Mixed	40.63 ± 5.94	80.00	29.53 ± 4.63	63.33	46.09 ± 2.09	84.85	57.01 ± 1.95	94.78
	ParoQ	Generic	39.12 ± 5.98	83.33	28.54 ± 5.13	70.00	46.34 ± 2.71	85.86	54.81 ± 3.19	98.26
	ParoQ++	OpenR1	38.86 ± 7.28	83.33	27.60 ± 4.66	66.67	46.46 ± 2.88	83.33	56.66 ± 2.77	96.96
	LAQuant	OpenR1	41.41 ± 5.90	83.33	31.77 ± 5.41	73.33	45.55 ± 2.88	86.87	58.67 ± 1.71	95.22

Table 14: Pass@1$^{\uparrow}$ on **MMLU-Pro** and **LiveCodeBench** under 3-bit and 4-bit quantization. R-QAT and ParoQ denote ReasoningQAT and ParoQuant, respectively. Results on LiveCodeBench benchmarks are averaged over three random seeds, 0, 1, and 42.

(a) W3G128 Pass@1 results.

Model	Method	Dataset	MMLU-Pro	LCB
Qwen3-1.7B	FP16		57.31	33.71
	GPTQ	Generic	22.34	0.00
	GPTQ++	OpenR1	35.74	3.73
	R-QAT	Mixed	47.48	19.90
	ParoQ	Generic	43.79	12.94
	ParoQ++	OpenR1	45.84	19.40
	LAQuant	OpenR1	47.94	21.02
Qwen3-4B	FP16		70.23	54.48
	GPTQ	Generic	53.32	2.99
	GPTQ++	OpenR1	60.61	29.97
	R-QAT	Mixed	63.24	37.31
	ParoQ	Generic	63.41	38.31
	ParoQ++	OpenR1	62.41	45.03
	LAQuant	OpenR1	62.63	45.15
Qwen3-8B	FP16		73.44	58.46
	GPTQ	Generic	65.71	37.94
	GPTQ++	OpenR1	66.29	46.64
	R-QAT	Mixed	68.33	44.16
	ParoQ	Generic	68.93	44.65
	ParoQ++	OpenR1	63.32	51.12
	LAQuant	OpenR1	66.86	51.74
R1-Distill-Llama-8B	FP16		58.84	36.32
	GPTQ	Generic	47.10	24.13
	GPTQ++	OpenR1	46.75	26.49
	R-QAT	Mixed	51.05	29.11
	ParoQ	Generic	44.29	1.12
	ParoQ++	OpenR1	51.44	29.47
	LAQuant	OpenR1	51.70	31.22

(b) W4G128 Pass@1 results.

Model	Method	Dataset	MMLU-Pro	LCB
Qwen3-1.7B	FP16		57.31	33.71
	GPTQ	Generic	50.49	20.40
	GPTQ++	OpenR1	54.21	28.48
	R-QAT	Mixed	54.44	25.87
	ParoQ	Generic	54.58	28.61
	ParoQ++	OpenR1	55.44	28.98
	LAQuant	OpenR1	55.58	31.84
Qwen3-4B	FP16		70.23	54.48
	GPTQ	Generic	68.03	48.75
	GPTQ++	OpenR1	68.00	50.75
	R-QAT	Mixed	68.28	46.76
	ParoQ	Generic	69.41	50.00
	ParoQ++	OpenR1	68.73	51.62
	LAQuant	OpenR1	68.41	52.74
Qwen3-8B	FP16		73.44	58.46
	GPTQ	Generic	72.67	52.99
	GPTQ++	OpenR1	72.33	55.10
	R-QAT	Mixed	72.65	51.62
	ParoQ	Generic	72.43	55.47
	ParoQ++	OpenR1	72.59	55.72
	LAQuant	OpenR1	71.52	56.59
R1-Distill-Llama-8B	FP16		58.84	36.32
	GPTQ	Generic	56.78	34.21
	GPTQ++	OpenR1	56.30	34.45
	R-QAT	Mixed	55.96	34.33
	ParoQ	Generic	56.69	33.58
	ParoQ++	OpenR1	56.77	33.58
	LAQuant	OpenR1	54.98	35.70

Table 15: Paired comparison of LAQuant and ParoQuant++. All differences are LAQuant minus ParoQuant++ in percentage points. W3 and W4 denote W3G128 and W4G128; 1.7B, 4B, and 8B denote Qwen3 models, and R1-8B denotes DeepSeek-R1-Distill-Llama-8B. A24, A25, and MMLU denote AIME24, AIME25, and MMLU-Pro.

(a) Six-benchmark average.

Config	Δ	95% CI	p
1.7B W3	+2.06	$[+0.68, +3.21]$	0.0022
4B W3	+0.92	$[-0.65, +1.80]$	0.354
8B W3	+2.18	$[+1.17, +3.02]$	0.0001
R1-8B W3	+1.81	$[+0.77, +2.73]$	0.0002
1.7B W4	+1.17	$[-0.56, +1.33]$	0.425
4B W4	+0.35	$[-0.73, +1.11]$	0.683
8B W4	+0.78	$[-0.10, +1.36]$	0.093
R1-8B W4	+1.05	$[-0.18, +1.99]$	0.109
Overall	+1.29	$[+0.69, +1.41]$	0.0001

(b) Task-level differences; bold indicates paired $p < 0.05$.

Config	AIME24	AIME25	GPQA	LSAT	MMLU	LCB
1.7B W3	+1.0	+1.1	+1.0	**+5.6**	**+2.1**	+1.6
4B W3	+3.2	+1.9	+0.0	+0.1	+0.2	+0.1
8B W3	+1.1	**+2.9**	+0.6	**+4.3**	**+3.5**	+0.6
R1-8B W3	+0.8	+3.0	**+2.9**	+2.1	+0.3	+1.8
1.7B W4	+0.5	+2.8	+0.1	+0.6	+0.1	+2.9
4B W4	+0.1	+0.9	+0.2	+0.2	-0.3	+1.1
8B W4	+0.4	**+2.7**	+0.6	+1.3	**-1.1**	+1.0
R1-8B W4	+2.6	**+4.2**	-0.9	+2.0	**-1.8**	+0.3

Paired Statistical Analysis of LAQuant and ParoQuant++. Both methods are evaluated on identical problem sets, allowing us to compare their scores within each problem. We use a paired cluster bootstrap over problems with 20,000 resamples to obtain confidence intervals and p-values for differences in the six-benchmark average Pass@1. Resampling paired observations preserves the correlation between methods and is particularly useful for the small, variable AIME benchmarks.

Table 15a shows that LAQuant improves the six-benchmark average in all eight model and quantization settings. Averaged across these settings, the gain is 1.29 percentage points (95% CI [0.69, 1.41]; p = 0.0001). Under W3G128, the paired gain is statistically significant for three of the four models: Qwen3-1.7B, Qwen3-8B, and R1-Distill-Llama-8B. The gains under W4G128 are smaller and do not reach significance in any individual model setting.

Table 15b provides the corresponding task-level differences. Of the ten cells marked significant at the reported p < 0.05 threshold, eight favor LAQuant. The two significant differences favoring ParoQuant++ occur on MMLU-Pro under W4G128. Thus, the aggregate improvement does not imply an advantage on every task: the strength of the evidence varies by model, bit width, and benchmark. Even where the W4G128 accuracy difference is not statistically significant, LAQuant avoids the online transformations required by ParoQuant++ and achieves higher decoding throughput in the evaluated settings.

G.2 General Tasks

Table 16: Detailed zero-shot accuracy$^{\uparrow}$ results on general tasks under 3-bit and 4-bit quantization.

Model	Method	Bits	ARC-C	ARC-E	BoolQ	Hella.	Avg.
Qwen3-1.7B	FP16	16	41.30	73.36	78.99	46.15	59.95
	GPTQ	3	33.28	66.96	62.69	43.94	51.72
	ParoQ	3	40.53	73.95	73.00	46.64	58.53
	LAQuant	3	39.16	72.39	76.51	46.37	58.61
	GPTQ	4	38.74	71.89	77.58	47.57	58.95
	ParoQ	4	41.98	74.87	77.46	48.67	60.75
	LAQuant	4	41.72	74.58	76.06	48.43	60.20
Qwen3-4B	FP16	16	48.12	79.17	83.24	54.57	66.28
	GPTQ	3	42.15	74.03	73.40	50.86	60.11
	ParoQ	3	49.23	80.55	81.59	52.67	66.01
	LAQuant	3	48.04	80.43	80.03	53.05	65.39
	GPTQ	4	48.46	78.83	81.74	54.11	65.79
	ParoQ	4	50.00	81.06	84.04	54.35	67.36
	LAQuant	4	48.38	79.59	83.45	54.33	66.44
Qwen3-8B	FP16	16	52.65	81.78	82.60	58.78	68.95
	GPTQ	3	50.94	82.24	80.49	55.69	67.34
	ParoQ	3	52.05	81.57	83.79	57.01	68.61
	LAQuant	3	53.50	82.62	84.22	57.07	69.35
	GPTQ	4	54.10	82.66	82.75	58.41	69.48
	ParoQ	4	52.39	82.66	84.50	58.57	69.32
	LAQuant	4	53.50	82.53	82.72	58.51	69.32
Llama3.1-8B	FP16	16	51.02	81.48	82.20	60.00	68.68
	GPTQ	3	44.28	76.85	74.89	56.29	63.08
	ParoQ	3	45.99	78.24	80.15	57.67	65.51
	LAQuant	3	45.99	78.37	80.61	57.57	65.64
	GPTQ	4	50.19	81.48	82.02	59.31	68.25
	ParoQ	4	50.09	81.14	81.59	59.65	68.12
	LAQuant	4	50.17	81.89	82.14	59.25	68.36

G.3 Decoding Throughput

Table 17: Decoding (batch size = 1) throughput (tokens/sec) across different GPU platforms.

DGX A100	Bits	Qwen3-1.7B	Qwen3-4B	Qwen3-8B	Llama3.1-8B
FP16	16	207.9	113.8	72.8	74.93
ParoQuant	4	217.4 (x1.05)	140.2 (x1.23)	116.4 (x1.60)	126.59 (x1.69)
LAQuant	4	260.8 (x1.25)	163.1 (x1.43)	131.4 (x1.81)	143.21 (x1.91)
ParoQuant	3	285.4 (x1.37)	214.9 (x1.89)	203.8 (x2.80)	233.73 (x3.12)
LAQuant	3	359.3 (x1.73)	274.9 (x2.42)	256.0 (x3.52)	294.37 (x3.93)
RTX 4090	**Bits**	**Qwen3-1.7B**	**Qwen3-4B**	**Qwen3-8B**	**Llama3.1-8B**
FP16	16	187.3	92.5	55.0	56.30
ParoQuant	4	271.6 (x1.45)	168.4 (x1.82)	114.6 (x2.08)	122.66 (x2.18)
LAQuant	4	302.1 (x1.61)	183.5 (x1.99)	121.8 (x2.21)	129.71 (x2.30)
ParoQuant	3	365.8 (x1.95)	279.6 (x3.02)	243.2 (x4.42)	282.94 (x5.30)
LAQuant	3	426.0 (x2.27)	322.1 (x3.48)	276.6 (x5.03)	322.11 (x5.72)
RTX 6000 Ada	**Bits**	**Qwen3-1.7B**	**Qwen3-4B**	**Qwen3-8B**	**Llama3.1-8B**
FP16	16	187.1	90.8	52.5	53.55
ParoQuant	4	279.7 (x1.49)	174.4 (x1.92)	112.6 (x2.14)	120.93 (x2.26)
LAQuant	4	313.9 (x1.68)	191.4 (x2.11)	122.5 (x2.33)	129.26 (x2.41)
ParoQuant	3	392.9 (x2.10)	278.2 (x3.06)	241.3 (x4.59)	298.15 (x5.57)
LAQuant	3	468.7 (x2.50)	325.0 (x3.58)	276.9 (x5.27)	345.47 (x6.45)

Problem. Let k be a real number such that the system

$$|25 + 20i - z| = 5,$$
$$|z - 4 - k| = |z - 3i - k|$$

has exactly one complex solution z. The sum of all possible values of k can be written as $\frac{m}{n}$, where m and n are relatively prime positive integers. Find $m + n$. Here $i = \sqrt{-1}$. Please reason step by step, and put your final answer within \boxed{}.

Correct Answer

4952 tokens

Let $z = x + yi$, where $x, y \in \mathbb{R}$. The first equation gives

$$|25 + 20i - z| = 5,$$

so

$$(x - 25)^2 + (y - 20)^2 = 25.$$

Thus, the first equation represents a circle centered at $(25, 20)$ with radius 5.
The second equation is

$$|z - 4 - k| = |z - 3i - k|.$$

Substituting $z = x + yi$,

$$|(x - 4 - k) + yi| = |(x - k) + (y - 3)i|.$$

Squaring both sides,

$$(x - 4 - k)^2 + y^2 = (x - k)^2 + (y - 3)^2.$$

Expanding and simplifying gives

$$-8x + 6y + 8k + 7 = 0,$$

or equivalently

$$4x - 3y = \frac{7}{2} + 4k.$$

Therefore, the second equation is a line.
The system has exactly one solution precisely when this line is tangent to the circle. Hence the distance from $(25, 20)$ to

$$4x - 3y - \frac{7}{2} - 4k = 0$$

must equal the radius 5. Thus,

$$\frac{|4(25) - 3(20) - \frac{7}{2} - 4k|}{\sqrt{4^2 + (-3)^2}} = 5.$$

Since $\sqrt{4^2 + (-3)^2} = 5$,

$$\frac{|\frac{73}{2} - 4k|}{5} = 5.$$

Therefore,

$$\left|\frac{73}{2} - 4k\right| = 25.$$

So

$$\frac{73}{2} - 4k = \pm 25.$$

This gives

$$k = \frac{23}{8} \quad \text{or} \quad k = \frac{123}{8}.$$

The sum of all possible values of k is

$$\frac{23}{8} + \frac{123}{8} = \frac{146}{8} = \frac{73}{4}.$$

Thus $m = 73$, $n = 4$, and

$$\boxed{77}$$

Incorrect Answer

32626 tokens

The first equation is interpreted correctly as

$$|z - (25 + 20i)| = 5,$$

so z lies on a circle centered at $(25, 20)$ with radius 5.
The second equation is

$$|z - 4 - k| = |z - 3i - k|.$$

It is rewritten as

$$|z - (4 + k)| = |z - (3i + k)|,$$

and then interpreted as the perpendicular bisector of the segment joining $4 + k$ and $3i + k$.
At this point, take the two points to be

$$A = (4, 0), \qquad B = (k, 3).$$

Then the midpoint is

$$M = \left(\frac{4 + k}{2}, \frac{3}{2}\right),$$

and the slope of AB is

$$\frac{3}{k - 4}.$$

Thus the slope of the perpendicular bisector is

$$-\frac{k - 4}{3}.$$

So the perpendicular bisector is

$$y - \frac{3}{2} = -\frac{k - 4}{3}\left(x - \frac{4 + k}{2}\right).$$

Putting this into standard form gives

$$2(k - 4)x + 6y + 7 - k^2 = 0.$$

For exactly one solution, this line must be tangent to the circle, so its distance from $(25, 20)$ is set equal to 5:

$$\frac{|2(k - 4)(25) + 6(20) + 7 - k^2|}{\sqrt{[2(k - 4)]^2 + 6^2}} = 5.$$

The numerator becomes

$$|-k^2 + 50k - 73| = |k^2 - 50k + 73|,$$

and the denominator becomes

$$2\sqrt{(k - 4)^2 + 9}.$$

Therefore,

$$\frac{|k^2 - 50k + 73|}{2\sqrt{(k - 4)^2 + 9}} = 5.$$

Squaring gives

$$(k^2 - 50k + 73)^2 = 100((k - 4)^2 + 9),$$

which expands to

$$k^4 - 100k^3 + 2546k^2 - 6500k + 2829 = 0.$$

This leads to an unnecessary quartic because the coordinate interpretation above is wrong: $4 + k$ is the real point $(k + 4, 0)$, not $(4, 0)$.

Incomplete generation due to verification repetition.

Figure 4: Comparison of generated solutions for AIME2025 Problem 7. The left panel shows the correct answer produced with LAQuant, while the right panel shows the incorrect ParoQuant output, which fails due to incomplete generation caused by verification repetition.

H Details of INT3 GEMV Kernel Modifications

The reference GPTQ INT3 GEMV kernel [13] is optimized for OPT-175B on A100 and 2×A6000 configurations, employing a fixed thread block size of BLOCKWIDTH $= 256$ and supporting only per-channel quantization. When applied to models in the Qwen3 and Llama families, this configuration yields suboptimal occupancy and lacks compatibility with both group-wise quantization and recent compilation infrastructure. We introduce three modifications to the kernel while preserving its original 3-bit packing format (32 weights encoded across three 32-bit words).

Templated thread block size. The reference kernel statically defines BLOCKWIDTH $= 256$ input features per thread block, with a corresponding BLOCKHEIGHT $= 24$ packed rows. For the matrix shapes encountered in small-to-medium models, this configuration produces an insufficient number of CUDA blocks to fully saturate modern GPUs. For instance, the q_{proj} layer of Qwen3-1.7B (2048×2048) issues a grid of only $8 \times 8 = 64$ blocks, corresponding to 0.6 blocks per streaming multiprocessor (SM) on the A100. To address this, we template the kernel on BLOCKWIDTH $\in \{64, 128, 256\}$, with BLOCKHEIGHT derived as BLOCKWIDTH$/32 \cdot 3$. For the matrix shapes in the Qwen3 family, BLOCKWIDTH $= 128$ increases the block count by a factor of four and consistently yields the highest throughput.

Group-wise quantization. The published GPTQ kernel supports only per-channel quantization, in which a single $(\textit{scale}, \textit{zero})$ pair is shared across an entire output column. We extend the kernel to support group-wise quantization with groupsize $= 128$ by storing the quantization parameters as a $(\textit{num_groups}, \textit{outfeatures})$ tensor and reloading them at the beginning of each 32-element inner-loop iteration. As the inner loop processes exactly 32 input elements per iteration and the group size is a multiple of 32, group boundaries are guaranteed to align with iteration boundaries, ensuring that the additional control flow incurs negligible runtime cost. This modification brings the kernel into alignment with prevailing quantization recipes and improves accuracy at equivalent bit-widths.

Custom-op registration. The reference kernel is exposed only through a pybind11 binding, which prevents `torch.compile` with `fullgraph=True` from tracing through it. We register the kernel as a `torch.library.custom_op` with a fake-tensor implementation, allowing the kernel to be captured into the compiled decode graph alongside the rest of the model.

I Additional Ablation Study

Table 18: Effect of training-token segment location on reasoning performance. Values denote Pass@$1^{\uparrow}$ on five reasoning benchmarks and their average.

Location	Bits	AIME25	GPQA	LSAT	MMLU-Pro	LCB	Avg.
	16	35.94	40.70	64.10	57.31	34.33	46.48
Prefix-2k	3	29.33	31.79	54.86	47.94	21.64	**37.11**
Response-2k	3	28.54	31.90	51.14	46.82	19.40	35.56
Middle-2k	3	29.79	31.50	50.98	47.02	19.40	35.74
Suffix-2k	3	26.88	31.09	53.86	46.82	20.52	35.83
Prefix-2k	4	35.47	36.43	61.14	55.98	32.46	**44.30**
Response-2k	4	34.48	36.40	60.68	55.07	30.97	43.52
Middle-2k	4	33.80	34.40	58.61	55.59	33.96	43.27
Suffix-2k	4	34.43	36.99	58.78	55.44	29.85	43.10

Training-Token Segment Location. We further examine whether the location of the training tokens within each chat-templated prompt–response sequence affects quantization performance, using a 2k-token segment, where 2k denotes 2,048 tokens. Specifically, *Prefix-2k* uses the first 2k tokens of the full sequence, *Response-2k* uses the first 2k tokens after `<im_start>assistant`, *Middle-2k* randomly samples a 2k-token span from the middle region after excluding the first and last 2k tokens, and *Suffix-2k* uses the last 2k tokens. As shown in Table 18, Prefix-2k achieves the highest average accuracy under both W3G128 and W4G128.

The results suggest that aligning the training segment with the token sequence observed during actual autoregressive decoding is important for reasoning-domain QAT. Although the response tokens

are the primary generation target, the prompt tokens remain in the KV cache throughout decoding and continuously participate in attention computation. Therefore, training on the prefix segment, which contains both the prompt and the early response context, better matches the inference-time computation than using only response-side or later tokens. This helps preserve the prompt-conditioned KV cache and provides more effective calibration signals for long-form reasoning generation.

Table 19: Ablation of trainable parameters on Qwen3-1.7B. All quantized variants use the one-layer lookahead loss ($k = 1$). Results are Pass@1; AIME25 and LSAT-AR are reported as mean $\pm$ standard deviation.

Setting	Trainable parameters	AIME25	LSAT-AR	LCB	Avg.
FP16	–	35.94 ± 4.58	64.10 ± 2.53	33.71	44.58
W3G128	QP only	11.72 ± 4.32	30.98 ± 1.98	8.45	17.05
	Weights only	21.72 ± 4.56	44.46 ± 2.07	12.62	26.27
	Both (LAQuant)	29.33 ± 4.87	54.86 ± 2.11	21.02	**35.07**
W4G128	QP only	30.00 ± 4.40	54.13 ± 2.18	24.75	36.29
	Weights only	32.81 ± 5.19	59.59 ± 2.17	25.99	39.46
	Both (LAQuant)	35.47 ± 4.54	61.14 ± 2.23	31.84	**42.82**

Contributions of Weight and Quantization-Parameter Training. Table 19 compares three choices of trainable parameters under the same one-layer lookahead objective: quantization parameters (QP) only, weights only, and both. Training weights alone outperforms QP-only training on all three benchmarks at both bit widths. The difference is especially pronounced under W3G128, where the three-benchmark average rises from 17.05 to 26.27. Jointly training weights and QP yields the highest accuracy in both settings, reaching 35.07 under W3G128 and 42.82 under W4G128. Thus, weight adaptation makes a substantial contribution under the lookahead objective, while QP updates provide a further benefit when combined with weight training.

J Additional Experiments

Scaling to Qwen3-14B. Table 20 extends our W3G128 reasoning evaluation beyond the 8B models in the main experiments. LAQuant achieves the highest three-benchmark average among the quantized methods, reaching 70.40% Pass@1 compared with 73.37% for FP16. It exceeds GPTQ++ by 4.87 percentage points and ParoQuant++ by 1.07 points on average. GPTQ++ performs slightly better on AIME25, while LAQuant leads the quantized methods on LSAT-AR and LiveCodeBench. These results show that LAQuant's ability to preserve reasoning accuracy extends to the 14B scale.

Table 20: Results on Qwen3-14B under W3G128 quantization. Values are Pass@1; AIME25 and LSAT-AR are reported as mean $\pm$ standard deviation.

Method	AIME25	LSAT-AR	LCB	Avg.
FP16	71.41 ± 4.85	88.01 ± 1.70	60.70	73.37
GPTQ++	66.67 ± 5.76	83.55 ± 1.82	46.39	65.54
ParoQuant++	64.37 ± 4.63	86.90 ± 1.77	56.71	69.33
LAQuant	**65.36** ± 4.68	**88.02** ± 1.63	**57.83**	**70.40**

Comparison with BRECQ and CBQ. Like our lookahead loss, BRECQ [31] and CBQ [9] extend reconstruction beyond a single layer: BRECQ reconstructs the output of each block, and CBQ jointly reconstructs a sliding window of consecutive blocks. Both aim to capture cross-layer dependencies of the quantization error from limited calibration data, whereas LAQuant targets the KV-cache fidelity that long autoregressive decoding depends on. We reproduce both methods on Qwen3-1.7B with the same calibration data and data budget as LAQuant. For BRECQ, we port the official implementation to decoder-only models, treating each decoder layer as a block and keeping its default AdaRound settings (20k iterations per block with an MSE reconstruction loss). For CBQ, we reimplement the

Table 21: Comparison with BRECQ and CBQ on Qwen3-1.7B (group size 128). Avg. is the mean of the three Pass@1 columns.

Bits	Method	AIME25		LSAT		LCB	Avg.
		Pass@1	Pass@64	Pass@1	Pass@16	Pass@1	
16	FP16	35.94 ± 4.58	73.33	64.10 ± 2.53	95.65	33.71	44.58
3	BRECQ	6.25 ± 2.92	23.33	35.48 ± 1.83	89.56	2.49	14.74
	CBQ	4.79 ± 4.11	30.00	23.37 ± 1.39	80.43	3.48	10.55
	LAQuant	29.33 ± 4.87	60.00	54.86 ± 2.11	93.91	21.02	**35.07**
4	BRECQ	33.96 ± 5.21	73.33	59.61 ± 2.20	92.17	27.49	40.35
	CBQ	31.25 ± 3.92	66.67	55.38 ± 2.17	91.30	27.49	38.04
	LAQuant	35.47 ± 4.54	73.33	61.14 ± 2.23	91.74	31.84	**42.82**

method from the paper with a two-block window, a one-block overlap, three epochs per window, and rank-5 LoRA-Rounding; we disable CFP weight truncation and apply hard rounding throughout optimization, since the paper's settings for both performed worse at group size 128. As shown in Table 21, LAQuant outperforms both at both bit widths. The gap is largest at W3G128, where BRECQ and CBQ collapse on AIME25 and LCB and reach averages of only 14.74 and 10.55, compared with 35.07 for LAQuant; at W4G128, LAQuant still leads (42.82 vs. 40.35 and 38.04).

Table 22: Effect of calibration data domain on Qwen3-1.7B (group size 128). LAQuant is trained on 4,096 samples of at most 2,048 tokens from either OpenR1-Math-220k (default) or OpenThoughts-114k-Code.

(a) Pass@1 (%); Avg. is the mean over the six benchmarks. Bold marks the better corpus at each bit width.

Bits	Calib. Data	AIME24	AIME25	GPQA	LSAT	MMLU-Pro	LCB	Avg.
16	–	45.84	35.94	40.70	64.10	57.31	33.71	46.27
3	OpenThoughts-Code	26.87	27.29	31.24	51.19	45.94	**25.50**	34.67
	OpenR1-Math	**34.64**	**29.33**	**31.79**	**54.86**	**47.94**	21.02	**36.60**
4	OpenThoughts-Code	40.26	33.07	34.44	60.24	54.72	**32.65**	42.56
	OpenR1-Math	**42.35**	**35.47**	**36.43**	**61.14**	**55.58**	31.84	**43.80**

(b) Hessian-subspace overlap (top-64 principal-angle cosine) between calibration corpora and evaluation traces.

Calib. Corpus	AIME25	GPQA	LSAT	LCB	Avg.
OpenR1-Math	0.893	0.680	0.651	0.702	0.732
OpenThoughts-Code	0.735	0.667	0.680	0.881	0.741
OpenThoughts3	0.728	0.692	0.681	0.829	0.733
ReasoningQAT mix	0.707	0.693	0.673	0.797	0.718
Generic (W2+C4+RP)	0.520	0.569	0.542	0.550	0.545

Effect of Calibration Data Domain. To test whether the Hessian-subspace analysis of Section 4.1 extends beyond math, we replace OpenR1-Math-220k with the decontaminated code subset of OpenThoughts-114k [17, 23], which contains DeepSeek-R1 traces on coding problems; all other settings follow the main results. Coding calibration raises LiveCodeBench Pass@1 by 4.48 pp at W3G128 and 0.81 pp at W4G128 but lowers every other benchmark, most sharply AIME24 at W3G128 (7.77 pp; Table 22a). The overlap anticipates this trade-off (Table 22b): OpenThoughts-Code aligns best with LiveCodeBench traces (0.881 vs. 0.702 for OpenR1-Math) and worse with AIME25 (0.735 vs. 0.893). When the gap is small (≤ 0.03 on GPQA and LSAT), the overlap does not reliably order the corpora; on LSAT, OpenR1-Math scores higher despite lower overlap. Given BF16 reasoning traces from a target domain, a large overlap gap can therefore guide the choice of calibration data before quantization.

K Licenses and Terms of Use for Existing Assets

Licenses for existing assets. Table 23 summarizes the licenses and terms of use for the existing assets used in this work, including models, training datasets, and evaluation benchmarks. We use these assets only for research purposes and follow the corresponding licenses, terms of use, and attribution requirements.

Table 23: Licenses and terms of use for existing assets used in this work.

Asset	Type	License / Terms of Use
WikiText-2	Dataset	CC BY-SA 4.0
C4	Dataset	ODC-BY; subject to Common Crawl Terms of Use
RedPajama	Dataset	Subset-dependent licenses; Common Crawl Terms of Use for web data
AIME24	Benchmark	Apache-2.0 for the dataset source used
AIME25	Benchmark	Apache-2.0 for the dataset source used
GPQA-Diamond	Benchmark	CC BY 4.0
LSAT-AR	Benchmark	MIT
LiveCodeBench	Benchmark	MIT for the benchmark toolkit; original platform terms apply
OpenR1-Math-220k	Dataset	Apache-2.0
OpenThoughts3	Dataset	Apache-2.0
MMLU-Pro	Benchmark	MIT
ARC-C / ARC-E	Benchmark	CC BY-SA 4.0
BoolQ	Benchmark	CC BY-SA 3.0
HellaSwag	Benchmark	MIT
Qwen3 series	Model	Apache-2.0
Llama-3.1-8B	Model	Llama 3.1 Community License
DeepSeek-R1-Distill-Llama-8B	Model	MIT; derived from Llama-3.1-8B-Base under the Llama 3.1 license

L Limitations and Future Work

LAQuant has several limitations worth acknowledging. At W3G128, a measurable gap to FP16 remains on competitive reasoning: 9.67, 8.28, and 5.53 pp on the Qwen3 1.7B/4B/8B reasoning average (Table 1). The gap shrinks with model scale, suggesting that low-bit reasoning quantization is hardest for smaller models, where each weight carries proportionally more task-relevant information.

Although Section 4.1 formalizes Hessian-subspace alignment as the diagnostic for matching calibration to deployment, the corpus we use (DeepSeek-R1 traces over OpenR1-Math-220k) was chosen empirically and validated post-hoc against the diagnostic. We do not provide a procedure for selecting or synthesizing calibration data given a target deployment distribution. The approach also implicitly assumes that a teacher trace generator (DeepSeek-R1 in our experiments) is available for the deployment domain; in domains lacking such a generator, calibration falls back to general text and the alignment guarantee weakens.

Larger open reasoning models (DeepSeek-R1-distilled 70B, Qwen3-30B-A3B, Qwen3-235B-A22B) exceed our compute budget. Layer-wise QAT cost scales linearly with parameter count, so the method is tractable in principle, but verifying that the gradient-direction analysis transfers at scale is left to future work. The lookahead loss itself adds a training-time memory cost: the teacher's ℓ and $\ell{+}1$ activations must be held simultaneously, raising peak VRAM relative to plain layer-wise QAT (Table 4).

Several extensions are natural follow-ups. Extending LAQuant to WbAbKVb regimes would compound the inference benefits and could augment the implicit Jacobian weighting with an explicit K- and V-cache reconstruction term that directly targets the cache-writing projections rather than relying on the implicit attribution alone. The principal-angle diagnostic of Section 4.1 could likewise be operationalized as a constructive procedure: for example, greedy maximization of $\bar{c}_k(\mathcal{D}_{\mathrm{cal}}, \mathcal{D}_{\mathrm{eval}})$ over a candidate corpus, or Hessian-aware prompt synthesis when no suitable corpus exists. Quantization of mixture-of-experts models (e.g., Qwen3-30B-A3B, Qwen3-235B-A22B) is another open direction: per-expert calibration coverage interacts non-trivially with the Hessian-alignment diagnostic, since each expert's activation distribution depends on the router's pattern under the calibration corpus. Finally, Table 5a reports $k{=}1$ as the best uniform lookahead depth, but layers vary in downstream-Jacobian conditioning; allowing k to vary per layer (e.g., based on a cheap spectral proxy of $J_{\ell+1}$) could recover the marginal gain that uniform $k{=}2$ does not deliver.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Throughout Section 3, 4, and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitation section is in Appendix L.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Propositions and lemmas are in sections 4.3 and 4.4. Their proofs are provided in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The information is provided in Section 5 and Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: For data, we only used open-source datasets. For code, we provided a GitHub URL in the abstract.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The information is provided in Section 5 and Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All reasoning benchmark results with std value are provided in the Appendix G.1. We also conducted paired comparisons in Table 15.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The training cost is provided in Section 5.2 and Table 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All authors have reviewed the code of ethics. There is little concern since only fully open-sourced and widely used models and datasets are used without any experiments with human subjects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is solely for improving the inference efficiency of recent large reasoning models.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: This work is applied only to fully open-source models and datasets. No new model nor dataset is proposed.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All models and datasets are properly referenced, especially in Section 5. The licenses and terms of use for the existing assets are explicitly summarized in Section K.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: This work only uses existing assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: No experiment with human subjects is conducted.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No experiment with human subjects is conducted.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLM usage is limited to polishing the manuscript (e.g., improving English, fixing typos, and figures & tables placement) and writing visualization codes that do not affect the core proposed methodology.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.