ParoQuant: Pairwise Rotation Quantization for Efficient Reasoning LLM Inference

Yesheng Liang[3,†] **Haisheng Chen**[3,‡] **Song Han**[1,2] **Zhijian Liu**[1,3]
[1]NVIDIA [2]MIT [3]UC San Diego
[†]Algorithm lead [‡]System lead

Abstract

Weight-only post-training quantization (PTQ) compresses the weights of Large Language Models (LLMs) into low-precision representations to reduce memory footprint and accelerate inference. However, the presence of outliers in weights and activations often lead to large quantization errors and severe accuracy degradation, especially in recent reasoning LLMs where errors accumulate across long chains of thought. Existing PTQ methods either fail to sufficiently suppress outliers or introduce significant overhead during inference. In this paper, we propose **Pairwise Rotation Quantization** (ParoQuant), a weight-only PTQ method that combines hardware-efficient and optimizable *Givens rotations* to even out the magnitude across channels and narrow the dynamic range within each quantization group. We further co-design the inference kernel to fully exploit GPU parallelism and keep the rotations and scaling lightweight at runtime. ParoQuant achieves an average **2.4%** accuracy improvement over AWQ on reasoning tasks with less than 10% overhead. This paves the way for more efficient and accurate deployment of reasoning LLMs.

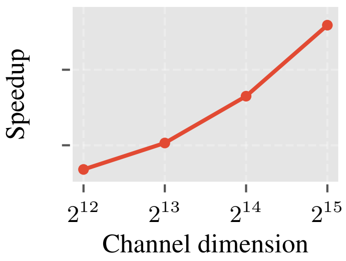
1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks. However, their massive size and large memory footprint not only incur substantial inference costs but also hinder on-device deployment. To address this, weight-only post-training quantization (PTQ) converts model weights to lower-bit-width representations (*e.g.*, INT4), reducing the memory footprint during inference and thus improving throughput in memory-bound autoregressive decoding.

Nevertheless, both activations and weights in LLMs possess many outliers (Dettmers et al., 2022; Xiao et al., 2023; Lin et al., 2024b), making it challenging to preserve the original precision under low-bit quantization. Most existing PTQ methods (Frantar et al., 2023; Lin et al., 2024b; Wei et al., 2023; Shao et al., 2024; Lee et al., 2024; Ashkboos et al., 2024; Chen et al., 2025; Tseng et al., 2024a;b) try to mitigate the impact of outliers, yet they either incur large quantization errors due to suboptimal outlier elimination or introduce significant overhead from arithmetic-intensive computation. For example, AWQ (Lin et al., 2024b), a widely adopted and fast quantization method, causes a **3.5% accuracy drop** of 4-bit quantized Qwen3-8B (Yang et al., 2025) on MMLU-Pro (Wang et al., 2024). In contrast, QTIP (Tseng et al., 2024b), which achieves state-of-the-art quantization accuracy, is about **30% slower** than AWQ because of the extra overhead introduced to mitigate outliers.

With the advent of reasoning LLMs (Jaech et al., 2024; Guo et al., 2025; Yang et al., 2025), we argue that *both* accuracy and efficiency are critical for practical quantization methods. Reasoning models achieve superior performance by generating a large number of chain-of-thought tokens, presenting unique challenges for quantization. On the one hand, quantization error *accumulates* at each decoding step, which becomes particularly pronounced in long generation. On the other hand, the substantial computational cost of generating long sequences requires that the quantization process itself introduce negligible overhead. Thus, there is a critical need for a quantization method that achieves high fidelity with minimal extra overhead.

In this paper, we propose **Pa**irwise **Ro**tation **Quant**ization (ParoQuant), a weight-only PTQ method that combines high accuracy with minimal computational overhead, making it well-suited to reasoning



5 Evaluation

Models and Tasks. We apply ParoQuant on LLaMA-2 (7B) (Touvron et al., 2023), LLaMA-3 (8B, 70B) & LLaMA-3.1 Instruct (8B) (Grattafiori et al., 2024), DeepSeek-R1-distilled LLaMA-3.1 (8B) (Guo et al., 2025), and Qwen3 (1.7B, 4B, 8B, 14B) (Yang et al., 2025) pre-trained models. We evaluate the quantization quality with three types of evaluation: (1) *Perplexity* on WikiText2 (Merity et al., 2017) and C4 (Dodge et al., 2021); (2) *Reasoning accuracy* on MMLU-Pro (Wang et al., 2024), GSM8K (Cobbe et al., 2021), GPQA Diamond (Rein et al., 2024), and AIME (MAA, 2024); (3) *Non-reasoning accuracy* on BoolQ (Clark et al., 2019), ARC-Challenge, ARC-Easy (Clark et al., 2018), and HellaSwag (Zellers et al., 2019).

Implementation. We focus on 4-bit weight-only linear quantization with a group size of 128. Linear quantization is more efficient and widely supported by existing frameworks. The choice of 4 bits and a 128 group size offers the optimal trade-off between accuracy and bit width for linear quantization (Dettmers & Zettlemoyer, 2023). We apply 8 independent rotations on each 128-channel group, with each rotation consisting of up to 64 pairs. Each layer is optimized for 10 epochs using AdamW (Loshchilov & Hutter, 2019) with a fixed set of hyperparameters for all experiments, except for the 70B model, where we adjust the batch size to accommodate memory constraints. To reduce the risk of overfitting to one dataset, we use a training set of 2048 samples drawn evenly from WikiText2, C4, and RedPajama (Weber et al., 2024), and select the best parameters at each training epoch using 64 validation samples from Pile (Gao et al., 2020). More details are provided in Section A.3.

Baselines. We compare the accuracy and efficiency of ParoQuant with three weight-only PTQ baselines. AWQ (Lin et al., 2024b) optimizes channel-wise scaling with grid search and is the most used 4-bit weight-only quantization method. EfficientQAT (Chen et al., 2025) achieves state-of-the-art linear quantization accuracy with layer-wise fine-tuning of weights and quantization parameters[*]. QTIP (Tseng et al., 2024b) is the state-of-the-art vector quantization method utilizing randomized Hadamard transform and an advanced trellis quantization algorithm. In addition, we include the perplexity results of QuIP# (Tseng et al., 2024a), a vector-quantization predecessor of QTIP that also adopts the Hadamard transform, and two weight-activation linear quantization methods, OmniQuant (Shao et al., 2024) and SpinQuant (Liu et al., 2025), which are also applicable for weight-only quantization. We apply block-wise quantization with a group size of 128 on all linear quantization methods and the corresponding default settings on vector quantization methods.

5.1 Accuracy Results

Perplexity. Table 1 shows the perplexity results of 4-bit quantized models ranging in size from 1.7B to 70B. Among linear quantization methods, ParoQuant achieves state-of-the-art quantization

[*] We only apply the "Block-AP" stage of EfficientQAT, as its "E2E-QP" stage involves supervised fine-tuning, which is out of the scope of PTQ.

accuracy across all sizes, particularly on challenging cases like LLaMA-3-8B and smaller models under 4B. It also delivers strong performance compared with rotation-based methods including QuIP#, QTIP, and SpinQuant. It outperforms QuIP# and matches QTIP on all models, despite the inherently larger error of linear quantization, highlighting the superior effectiveness of our proposed transform over the Hadamard transform (see Section A.2 for detailed analysis). Moreover, ParoQuant provides a decent speedup over these two methods. This underscores the efficiency of our proposed transform.

Method	Type	WikiText2							C4							Speedup
		L3-8	L3-70	L2-7	Q3-1.7	Q3-4	Q3-8	Q3-14	L3-8	L3-70	L2-7	Q3-1.7	Q3-4	Q3-8	Q3-14	
FP16	–	5.54	2.56	5.12	8.32	7.01	6.24	5.70	7.10	5.78	6.63	8.62	7.61	6.97	6.54	1.0×
QUIP#	vector	5.81	2.99	5.19	–	–	–	–	7.32	5.96	6.75	–	–	–	–	1.9×
QTIP	vector	5.67	2.75	5.17	–	–	–	–	7.20	5.83	6.69	–	–	–	–	1.7×
AWQ	linear	5.92	2.96	5.23	8.80	7.36	6.45	5.85	7.42	5.91	6.80	9.01	7.89	7.14	6.65	2.4×
OMNIQ	linear	–	–	5.23	–	–	–	–	–	–	6.80	–	–	–	–	2.4×†
SPINQ	linear	5.83	–	5.21	–	–	–	–	7.41	–	6.86	–	–	–	–	2.4×†
E-QAT	linear	5.87	3.33	5.22	8.60	7.19	6.37	5.82	7.36	6.72	6.76	8.84	7.77	7.08	6.63	2.4×†
PAROQ	linear	**5.72**	**2.82**	**5.18**	**8.43**	**7.10**	**6.30**	**5.75**	**7.26**	**5.86**	**6.74**	**8.75**	**7.70**	**7.04**	**6.60**	2.2×

† Uses results of AWQ as a reference as the method does not incur significant overhead from the transform.

Table 1: Perplexity (↓) results of 4-bit models. The context length is 8192 for LLaMA-3 and Qwen3 (base models), and 4096 for LLaMA-2. The best results among linear quantization methods are in **bold**. Speedup over FP16 models is reported as the geometric mean across Q3-1.7, Q3-4, L3-8, and Q3-14, measured on an RTX A6000 with a batch size of 1 during decoding.

Reasoning Tasks. Table 2 shows the accuracy results of five reasoning benchmarks: MMLU-Pro (12k samples), GSM8K (1.3k samples), GPQA Diamond (198 samples), and AIME24/25 (30 samples each). On the larger MMLU-Pro benchmark, ParoQuant consistently outperforms all baselines. While results on the smaller GSM8K, GPQA and AIME benchmarks exhibit more randomness due to the limited number of samples, ParoQuant's performance never degrades by more than 1.7%, showcasing a level of stability not seen in other baselines. Overall, ParoQuant preserves the reasoning capabilities of the original models and achieves **2.4%** and **2.9%** improvements over AWQ and EfficientQAT, respectively. This demonstrates ParoQuant's superior quantization accuracy in long generation.

Method	Type	R1-Distill-Llama-8B			Qwen3-8B				Qwen3-14B				Avg.
		GSM8K	GPQA	AIME	MMLU	GSM8K	GPQA	AIME	MMLU	GSM8K	GPQA	AIME	
FP16	–	72.9	39.4	38.3	68.6	73.5	44.4	70.0	72.2	88.2	53.0	76.7	63.4
QTIP	vector	71.0	37.9	40.0	–	–	–	–	–	–	–	–	–
AWQ	linear	70.3	40.4	35.0	65.1	71.4	**48.5**	**71.7**	70.9	87.2	52.0	73.3	62.3
E-QAT	linear	69.5	40.4	**43.3**	61.1	**79.8**	47.2	55.0	71.0	91.2	51.8	70.0	61.8
PAROQ	linear	**71.4**	**42.4**	**43.3**	**67.8**	75.9	47.5	**71.7**	**71.5**	**92.3**	**52.5**	**75.0**	**64.7**

Table 2: Accuracy (↑) on reasoning tasks. We report 5-shot accuracy for GSM8K and zero-shot accuracy for the other benchmarks. DeepSeek-R1-Distill-Llama-8B fails to produce reasonable results on MMLU within the set token limits. More details are provided in Section A.6.

Non-Reasoning Tasks. Table 3 shows the zero-shot accuracy on commonsense benchmarks with thinking mode disabled. ParoQuant maintains near-lossless performance, outperforming AWQ and EfficientQAT by 1% and 0.7%, respectively. The accuracy gap is smaller than in reasoning tasks because these benchmarks evaluate only a few generated tokens, so error accumulation is minimal.

5.2 Efficiency Results

Table 4 shows the decoding throughput on an RTX A6000. To ensure a fair comparison, we implement all methods on top of the Transformers library (Wolf et al., 2020), modifying only the weight transform and dequantization code (details and more results are in Section A.4). ParoQuant is only about 10% slower than AWQ while providing a significant accuracy improvement, and it matches the accuracy of QTIP while being 15%-30% faster. For the training efficiency, see Section A.5 for more details.

Method	Type	LLaMA-3.1-8B-Instruct				Qwen3-4B				Qwen3-8B				Qwen3-14B				Avg.
		BoolQ	ARC-C	ARC-E	HSwag	BoolQ	ARC-C	ARC-E	HSwag	BoolQ	ARC-C	ARC-E	HSwag	BoolQ	ARC-C	ARC-E	HSwag	
FP16	-	84.1	51.7	81.8	59.1	85.1	50.7	80.5	52.3	86.6	55.8	83.6	57.1	89.3	58.6	84.1	61.0	70.1
QTIP	vector	**84.3**	51.2	**81.7**	**58.8**	–	–	–	–	–	–	–	–	–	–	–	–	–
AWQ	linear	83.5	51.5	80.6	58.4	85.1	47.4	77.9	51.3	86.2	53.8	82.2	56.3	89.0	57.9	83.2	60.3	69.0
E-QAT	linear	83.8	**51.8**	81.6	58.4	**85.2**	47.2	78.4	51.2	**86.8**	54.7	82.7	**56.5**	88.8	58.1	83.7	60.3	69.3
ParoQ	linear	83.8	51.3	81.3	**58.8**	84.8	**51.0**	**80.4**	**51.5**	**86.8**	**56.3**	**84.1**	56.4	**89.6**	**58.6**	**84.3**	**60.7**	**70.0**

Table 3: Zero-shot accuracy (↑) on non-reasoning tasks.

Method	Qwen3-1.7B		Qwen3-4B		LLaMA-3-8B		Qwen3-14B	
	Throughput	W2 PPL	Throughput	W2 PPL	Throughput	W2 PPL	Throughput	W2 PPL
FP16	170 (1.0×)	8.32	78 (1.0×)	7.01	45 (1.0×)	5.54	25 (1.0×)	5.70
AWQ	320 (1.9×)	8.80	176 (2.3×)	7.36	120 (2.7×)	5.92	70 (2.8×)	5.85
QTIP	209 (1.2×)	–	117 (1.5×)	–	95 (2.1×)	5.67	55 (2.2×)	–
ParoQ	278 (1.6×)	8.43	160 (2.1×)	7.10	112 (2.5×)	5.72	65 (2.6×)	5.75

Table 4: Decoding (with batch size of 1) throughput (tokens/s).

6 Conclusion

In this paper, we proposed ParoQuant, an efficient weight-only PTQ method that achieves state-of-the-art quantization accuracy with minimal overhead. Based on the insight that a sparsely parameterized rotation can effectively suppress weight outliers, we designed scaled pairwise rotation, which combines hardware-friendly Givens rotations. ParoQuant matches the accuracy of the best existing quantization methods while running much faster, and it consistently outperforms prior efficient quantization methods, especially on reasoning tasks where quantization errors accumulate over long chains of thought. We hope that our method will inspire future research on high-fidelity, low-overhead quantization techniques for next-generation reasoning LLMs.