The Impact of Quantization on Large Reasoning Model Reinforcement Learning

Medha Kumar
Pennsylvania State University*
University Park, PA
mkumar@psu.edu

Zifei Xu
d-Matrix
Santa Clara, CA
xuzifei@d-matrix.ai

Xin Wang
d-Matrix
Santa Clara, CA
poincare.disk@gmail.com

Tristan Webb
d-Matrix
Santa Clara, CA
twebb@d-matrix.ai

Abstract

Strong reasoning capabilities can now be achieved by large-scale reinforcement learning (RL) without any supervised fine-tuning. Although post-training quantization (PTQ) and quantization-aware training (QAT) are well studied in the context of fine-tuning, how quantization impacts RL in large reasoning models (LRMs) remains an open question. To answer this question, we conducted systematic experiments and discovered a significant gap in reasoning performance on mathematical benchmarks between post-RL quantized models and their quantization-aware RL optimized counterparts. Our findings suggest that quantization-aware RL training negatively impacted the learning process, whereas PTQ and QLoRA led to greater performance.

1 Introduction

Large reasoning models are trained in data and algorithmic pipelines. Commonly, LLMs are first "pre-trained" on trillions of input tokens, to develop a strong general ability to model the distribution of the training data, as well as showing "sparks. . . of intellegence" [Bubeck et al., 2023]. Post pre-training, models undergo further fine tuning, and one popular technique pioneered by Shao et al. [2024] is to apply reinforcement learning to have the LLMs solve problems in domains with verifiable rewards, such as math or programming.

Quantization is a widespread technique for improving LLM memory and compute efficiency. As of 2025, new LLM "base" models commonly are released in FP16 or BF16 precision. Quantization is often left to fine tuners or software framework maintainers downstream from a model's official release. The intersection of RL and highly specialized agentic AI may lead to situations where many different LRM agents are derived from the same full precision base model, but specialized to different tasks through RL, and at some point quantized for inference performance. We study the question: how do we perform quantization to ensure the best test-time memory/performance tradeoff?

Prior work from authors Krishnan et al. [2019] found that during distributed training, the quantized actors could save energy. We are not aware of any other work examining the effect of quantization of training large reasoning models.

*Work done while interning at d-Matrix

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

2 Methods

Our main investigation was to evaluate the reasoning performance of LLM models under different quantization strategies, and to explore the trade-offs between strategies in practice. Our general setup is that we have been provided with a base LLM which will be fine tuned through reinforcement learning on a specialized downstream tasks (such as mathematics) to produce a LLM with enhanced reasoning, or in other words a large reasoning model (LRM). Broadly speaking, a practitioner can choose from quantization strategies that can be divided into a number of different categories: such as post training quantization (PTQ), quantization aware training/fine-tuning (QAT/QAFT), and low-rank adapter fine tuning techniques, such as QLoRA. After quantization to a selected precision, we evaluate the models on a test dataset. We have released our training and evaluation code online at `github.com/d-matrix-ai/rlquant`.

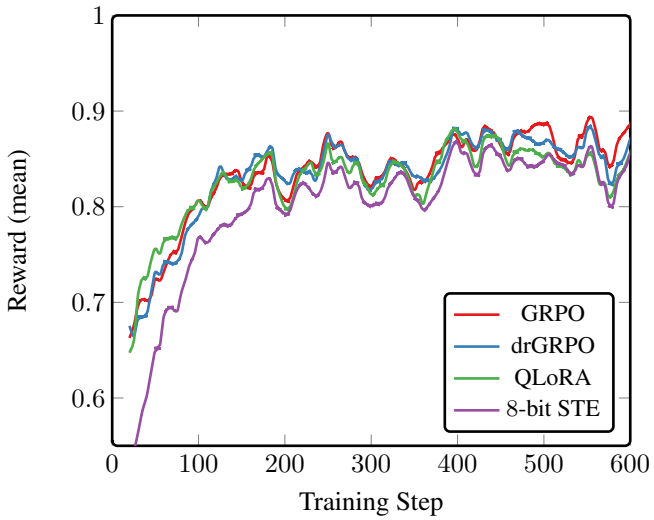

Figure 1: Mean training reward observed during RL training of Qwen3-8B, windowed moving average (window size $= 25$) shown.

2.1 Verifiable Reward RL training

We utilized both the GRPO [Shao et al., 2024] and drGRPO [Liu et al., 2025] algorithms to fine-tune base models from the Qwen3 [Yang et al., 2025] family of models on a variety of math benchmarks. We trained on the same MATH [Hendrycks et al., 2021] level 3-5 questions used by authors Liu et al. [2025], and evaluated on questions from AIME2024, AMC, MATH500, Minerva Math and OlympiadBench on data also open sourced by the the same authors. Simulations were run using the GRPO and drGRPO training from the `TRL` library. Simulations were trained on $10{,}000$ samples from the MATH dataset for 1 epoch, with a learning rate of 10^{-6}. In Fig. 1 we show the training reward over the course of the run. Our reward function assigned a reward of 1 for responses from the LLM that were mathematically correct, and furthermore added a reward of 0.1 to responses that provided correct output formatting for the reward, which means rewards would be sampled from the set $\{0, 0.1, 1.1\}$.

2.1.1 QAFT with 8-bit STE

One of the simplest methods to perform quantization aware fine tuning is before calculating activation to preform a Round to Nearest (RTN) quantization to the weight and then perform straight through estimation (STE) of the quantization gradients to avoid any non-differentiable sections of the graph. In our experiments, we quantized all of the linear layer weights (not activations) in the attention blocks to INT8.

Model	Qwen			
	0.6B	1.7B	4B	8B
(Base)	0.164	0.212	0.451	0.473
(GRPO)	**0.307**	**0.418**	**0.555**	**0.594**
(drGRPO)	0.287	0.389	0.541	0.584
(STE 8-bit)	**0.242**	0.325	0.443	0.496
(PTQ BnB 8-bit)	0.222	**0.366**	**0.528**	0.579
(PTQ AWQ 8-bit)	0.22	0.364	0.526	**0.583**
(QLoRA 4-bit)	**0.24**	**0.382**	**0.554**	0.556
(PTQ BnB 4-bit)	0.223	0.369	0.527	**0.581**
(PTQ AWQ 4-bit)	0.225	0.366	0.533	0.574

Table 1: Evaluation mean reward (rounded to 3 decimal places). (Base): full precision official Qwen3-Base models; (GRPO, drGRPO): full precision RL training; (STE): INT8 RTN, straight-through-estimator RL training; (PTQ): each method was performed on the full precision GRPO checkpoint evaluated above it.

2.1.2 QLoRA

Another quantization that is used during the RL training process is QLoRA [Dettmers et al., 2023]. This method is considered "parameter-efficient" since it introduces low-rank adapter matrices and during the RL process only the parameters in the smaller adapter matrices are updated while keeping the quantized weights frozen. QLoRA training was accomplished using the `PEFT` module from HuggingFace, and `bitsandbytes` (BnB) for quantization to NF4. QLoRA training used a learning rate of 10^{-4}, a rank of 8, and $\alpha = 16$. The higher learning rate was required for the model to learn despite the quantization noise. QLoRA utilizes full numerical precision during model training, and the low rank adapter matrices can be merged into the base model resulting in a quantized model.

2.2 PTQ via AWQ and bitsandbytes

There are numerous PTQ techniques a modern practitioner could choose from to quantize a LRM after it has been fine-tuned. Beyond the other PTQ approaches we studied, there exist many other accessible approaches, such GPTQ [Frantar et al., 2022], SpinQuant [Liu et al., 2024], GGUF [Gerganov, 2023], and many others. We chose two approaches that capture two different flavors of PTQ: data-free approaches, and those that use data to calibrate. With that, we selected the `bitsandbytes` and AWQ [Lin et al., 2023] to produce PTQ models at both 8 and 4-bit precision. We applied PTQ directly to the same GRPO checkpoints we show evaluation results for in Table 1. With `bitsandbytes` we specified a HuggingFace Quantization config to load the bit precision, and used the NF4 data type for 4-bit quantization. We used the AWQ implementation from `llmcompressor` [AI and vLLM Project, 2024].

3 Results

Our main results are shown in Table 1. We found that quantizing the network to 8-bit precision through QAFT style STE training results in the greatest quantization error in networks larger than 0.6B. We found the two PTQ techniques we examined performed well even at 4-bits. Overall, using 4-bit QLoRA during reinforcement learning training resulted in networks with the lowest quantization error in almost all cases.

3.1 Impact of completion length on model performance

For the 0.6B and 1.7B models we set a completion length (for both training and evaluation) of 512 tokens to obtain the evaluation scores shown in the results table. However, this completion length caused sub-optimal performance for the 4B and the 8B models. Table 2 shows the mean evaluation reward at varying completion lengths for 4B and 8B. Increasing the completion length helped both models learn more from the same number of training steps.

Model	Qwen	
	4B	8B
(GRPO @ 1024 tokens)	0.555	0.594
(GRPO @ 512 tokens)	0.487	0.540

Table 2: Impact of token length on GRPO fine-tuned model performance. The same length is used in both training and evaluation.

3.2 Evaluation score versus model size

Figure 2 shows a plot of the performance/memory trade-off for different models we evaluated, across sizes, RL training, and quantization. Our analysis shows that quantization generally offers stronger performance than using smaller full precision networks.

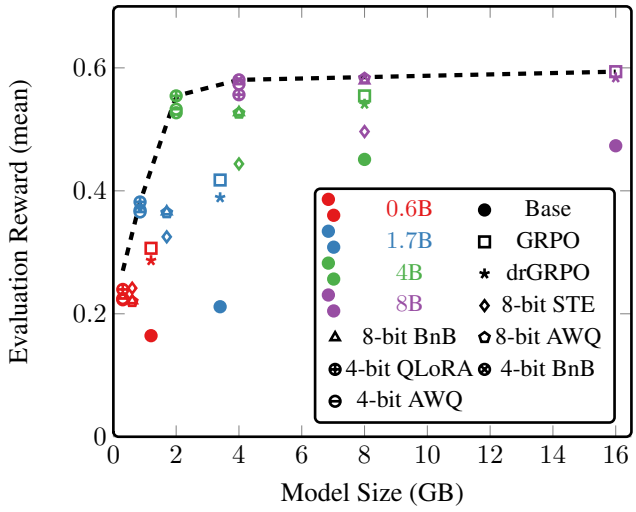

Figure 2: Evaluation reward vs. model size across all the models that we evaluated. We show the optimum pareto frontier as a dashed line.

4 Discussion

Our results show a strong trend across different model sizes. Techniques such as QAFT have generally been a sample efficient method[Xu et al., 2024] to quantize neural networks and maintain performance. However, techniques like reinforcement learning produce a unique challenge for the quantization of LLMs, in that a discrete rewards are sampled from LLM generated responses, and quantized models produce worse policies. Our results show that techniques that quantize models downstream from training such as PTQ and QLoRA result in models that reason better on downstream tasks and result in better performance/memory trade offs. We find these techniques very effective at preserving reasoning ability, even at 4-bit precision.

Our study should not be interpreted discouraging the use of QAT during large reasoning models pre-training. Rather, we show that a sudden shock of quantization during the reinforcement learning process is damaging to learning. If QAT/QAFT was initiated prior to reinforcement learning training, perhaps the model would have already adapted to the lower bit-precision and would have learned effectively. We leave this and the discovery of more efficient quantization techniques for large reasoning models to future work, and present this work as a guide for the modern practitioner.

References

R. H. AI and vLLM Project. LLM Compressor, 8 2024. URL https://github.com/vllm-project/llm-compressor.

S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36:10088–10115, 2023.

E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

G. Gerganov. llama.cpp: Port of Meta's Llama model in C/C++. https://github.com/ggerganov/llama.cpp, 2023. Accessed: [Date of Access, e.g., August 24, 2025].

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021.

S. Krishnan, M. Lam, S. Chitlangia, Z. Wan, G. Barth-Maron, A. Faust, and V. J. Reddi. QuaRL: Quantization for fast and environmentally sustainable reinforcement learning. *arXiv preprint arXiv:1910.01055*, 2019.

J. Lin, J. Tang, H. Tang, S. Yang, X. Dang, and S. Han. AWQ: Activation-aware weight quantization for LLM compression and acceleration. arxiv 2023. *arXiv preprint arXiv:2306.00978*, 2023.

Z. Liu, C. Zhao, I. Fedorov, B. Soran, D. Choudhary, R. Krishnamoorthi, V. Chandra, Y. Tian, and T. Blankevoort. SpinQuant: LLM quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024.

Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin. Understanding R1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Z. Xu, S. Sharify, T. Webb, X. Wang, et al. Understanding the difficulty of low-precision post-training quantization for llms. *arXiv preprint arXiv:2410.14570*, 2024.

A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.