

RoSTE: An Efficient Quantization-Aware Supervised Fine-Tuning Approach for Large Language Models

Quan Wei^{1*}, **Chung-Yiu Yau**^{2*}, Hoi-To Wai², Yang (Katie) Zhao¹,
Dongyeop Kang¹, Youngsuk Park³, Mingyi Hong¹

¹University of Minnesota, ²The Chinese University of Hong Kong,
³Amazon Web Services



INFORMS International 2025 @ Singapore

Background — Neural Network Quantization

- Modern deep learning models are super high-dimensional and memory consuming.
- While large model trained on high precision is accurate and present exceptional generalization ability, they are practically expensive to use. For example, a typical 8× H100 / A100 server has 640 GB GPU VRAM in total.

Model Size	FP16	INT4
8B	16 GB	4 GB
70B	140 GB	35 GB
405B	810 GB	203 GB

Table 1: Llama 3.1 GPU VRAM requirement for loading the model weights over different model sizes and weight precisions. (<https://huggingface.co/blog/llama31>)

- It calls for the development of **model quantization** that represent neural networks using low precision data-types such that INT4 and preserve the accuracy of prediction.

Quantization-Aware Training (QAT) Formulation

To train a neural network with low-precision data-type weights, we solve the following *Quantization-Aware Training* (QAT) problem with a stochastic training objective function $f(\mathbf{w}, \xi) : \mathbb{R}^d \times \Omega \rightarrow \mathbb{R} \setminus \{-\infty\}$:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \mathbb{E}_{\xi} [f(Q(\mathbf{w}), \xi)] \quad (1)$$

for a quantization function $Q(\mathbf{w}) = \text{Decode}(\text{Encode}(\mathbf{w}))$, $\text{Encode} : \mathbb{R}^d \rightarrow \mathbb{N}^d \times \mathbb{R}$, $\text{Decode} : \mathbb{N}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ that enables **memory efficient representation**, such as blockwise uniform symmetric quantization: by separating \mathbf{w} into n blocks $\mathbf{w} = [\mathbf{w}_{[1]}, \dots, \mathbf{w}_{[n]}]$,

$$Q(\mathbf{w}_{[i]}) = \underbrace{\text{clamp}_b \left(\left\lfloor \frac{\mathbf{w}_{[i]}}{s(\mathbf{w}_{[i]})} \right\rfloor \right)}_{b\text{-bits integer tensor}} s(\mathbf{w}_{[i]}) \quad (2)$$

where $s(\mathbf{w}_{[i]}) = \max(|\mathbf{w}_{[i]}|)/(2^{b-1} - 1)$ scales a block of weight values into the b -bits-representable integer range.

Existing QAT Algorithm — Straight-through Estimator (STE)

- [Courbariaux et al. 2015; Liu et al. 2023] solve the above problem with *straight-through estimated* (STE) stochastic gradient:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \nabla f(\mathbf{w}, \xi^t)|_{\mathbf{w}=Q(\mathbf{w}^t)} \quad (3)$$

where STE approximates the gradient of a non-differentiable quantization function Q by

$$\frac{\partial Q(w_i)}{\partial w_i} \approx 1 \quad (4)$$

- [Liu et al. 2023] successfully applied STE to train weight-activation quantized LLMs when quantization error $Q(\mathbf{w}_{[i]}) - \mathbf{w}_{[i]}$ is small.

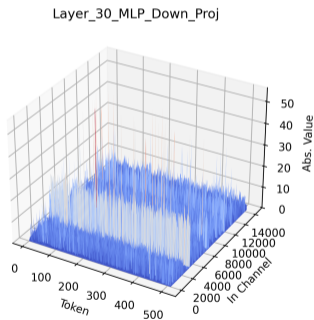
Our Problem — Quantizing Fine-tuned LLM

- **Supervised fine-tuning** adapts pre-trained LLMs to downstream tasks.
- Prior works perform **quantization** after training for efficient LLM deployment.
- To obtain quantized fine-tuned LLMs, conventional pipelines would first fine-tune the pre-trained models, followed by post-training quantization.

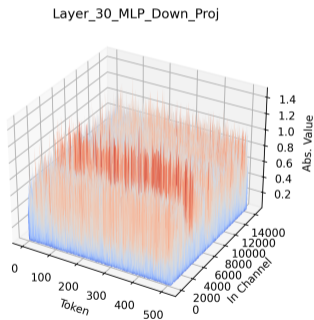
We investigate **quantization-aware supervised fine-tuning (QA-SFT)** to obtain effective fine-tuned and quantized LLM through a single training phase.

Main Issue — Large Quantization Error due to Outlier Values

- **Outlier** values in weight and activation leads to large error in **4-bits** uniform quantizer.
- **Rotation-based** methods (Ashkboos et al. 2024) apply rotations to linear projection layers and KV caches in LLMs effectively mitigates weight and activation outliers.



(a) Without rotation \mathbf{X} .



(b) With rotation \mathbf{RX} .

Figure 1: Visualizations of input activations \mathbf{X} (resp. \mathbf{RX}) at layer 30 of Llama model.

Rotation Matrices yield Accurate Quantized Linear Projection

Suppose $\mathbf{R}(\zeta) = \mathbf{H}\text{Diag}(\mathbf{r}(\zeta))$ where \mathbf{H} is a Hadamard rotation matrix and $\mathbf{r}(\zeta) \in \{-1, 1\}^d$ is a random sign vector. Then, we have $\mathbf{R}(\zeta)^\top = \mathbf{R}(\zeta)^{-1}$ and

$$Q(\mathbf{R}(\zeta)\mathbf{w})^\top Q(\mathbf{R}(\zeta)\mathbf{x}) \approx \mathbf{w}^\top \mathbf{x} \quad \text{when} \quad Q(\mathbf{R}(\zeta)\mathbf{w}) \approx \mathbf{R}(\zeta)\mathbf{w}, \quad Q(\mathbf{R}(\zeta)\mathbf{x}) \approx \mathbf{R}(\zeta)\mathbf{x} \quad (5)$$

Consider a b_w -bits symmetric quantizer Q_w , we compare

- Quantization error without rotation:

$$\|Q_w(\mathbf{w}) - \mathbf{w}\|^2 \leq \frac{d \max_i \mathbf{w}_i^2}{4(2^{b_w-1} - 1)^2}. \quad (6)$$

- Quantization error with rotation (Tseng et al. 2024): with high probability,

$$\|Q_w(\mathbf{R}(\zeta)\mathbf{w}) - \mathbf{R}(\zeta)\mathbf{w}\|^2 \leq \frac{\log(4d/\delta)}{2(2^{b_w-1} - 1)^2} \|\mathbf{w}\|^2. \quad (7)$$

- The former bound is **more sensitive** to weight outliers, i.e., $\max_i \mathbf{w}_i \gg \|\mathbf{w}\|^2$.

Rotation in LLM Modules

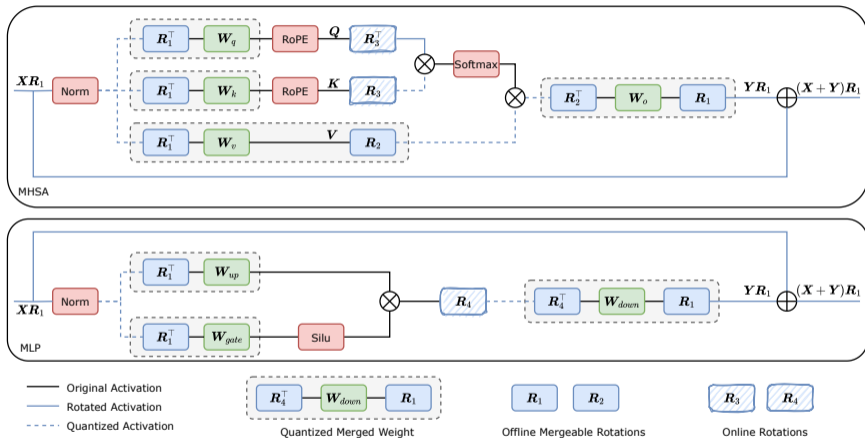


Figure 2: An illustration of the rotation workflow in a transformer-based model. R_1 represents the between-block rotation, which eliminates activation outliers between blocks. R_2 , R_3 , R_4 are in-block rotations designed to remove outliers within the MHSA and MLP blocks.

Proposed Algorithm: RoSTE

- **RoSTE: Rotation-based quantization for QA-SFT using Straight-Through Estimator.**
 - We propose an **adaptive selection** of rotation matrices during training.
- We utilize a **bilevel** optimization formulation that simultaneously tackles QA-SFT and selects the rotation matrices based on the current weights and activations:

$$\begin{aligned} \min_{\{\mathbf{W}_i\}_{i=0}^{\ell-1}} \mathcal{L}_{\text{SFT}}(Q(\cdot; \{\mathbf{W}_i, \mathbf{R}_i^*\}_{i=0}^{\ell-1})) \\ \text{s.t. } \{\mathbf{R}_i^*\}_{i=0}^{\ell-1} \in \arg \min_{\{\mathbf{R}_i\}_{i=0}^{\ell-1}} \mathcal{E}(\{\mathbf{W}_i, \mathbf{R}_i\}_{i=0}^{\ell-1}) \quad \text{s.t. } \mathbf{R}_i \mathbf{R}_i^\top = \mathbf{I}, \end{aligned}$$

where the lower level optimal rotation matrices $\{\mathbf{R}_i^*\}_{i=0}^{\ell-1}$ minimize the weight-activation quantization error:

$$\mathcal{E}(\{\mathbf{W}_i, \mathbf{R}_i\}_{i=0}^{\ell-1}) = \sum_{i=0}^{\ell-1} \|Q_w(\mathbf{R}_i^\top \mathbf{W}_i) - \mathbf{R}_i^\top \mathbf{W}_i\|^2 + \frac{1}{n} \sum_{i=0}^{\ell-1} \sum_{j=0}^{n-1} \|Q_x(\mathbf{X}_{i,j} \mathbf{R}_i) - \mathbf{X}_{i,j} \mathbf{R}_i\|^2$$

Proposed Algorithm: RoSTE

Input: Pre-trained model parameters $\{\mathbf{W}_i^{\text{pt}}\}_{i=0}^{\ell-1}$, step size $\eta > 0$.

Initialize: $\mathcal{W}^0 = \{\mathbf{W}_i^{\text{pt}}\}_{i=0}^{\ell-1}$.

for $k = 0, \dots, K - 1$ **do**

/* Rotation configuration */

Find an approximate lower level solution

$$\mathcal{R}^k = \arg \min_{\mathbf{R}_i \in \{\mathbf{H}, \mathbf{I}\}} \mathcal{E}(\mathcal{W}^{kT}, \{\mathbf{R}_i\}_{i=0}^{\ell-1}), \quad (8)$$

for identity matrix \mathbf{I} or random Walsh-Hadamard matrix \mathbf{H} .

for $t = 0, \dots, T - 1$ **do**

/* QAT Stage via STE */

$$\mathcal{W}^{kT+t+1} = \mathcal{W}^{kT+t} - \eta \overset{\text{s.t.e.}}{\nabla}_{\mathcal{W}} \mathcal{L}_{\text{SFT}}(m_Q(\cdot; \mathcal{W}^{kT+t}, \mathcal{R}^k); \xi^{kT+t}) \quad (9)$$

Output: Quantized fine-tuned $m_Q(\cdot; \mathcal{W}^{KT}, \mathcal{R}^{K-1})$.

Theoretical Insights of RoSTE

Consider a simple linear regression problem with quantized linear model:

$$\hat{\mathcal{L}}(\mathbf{w}, \mathbf{R}) := \frac{1}{2} \mathbb{E}_{\xi} \left[(Q_x(\mathbf{R}\mathbf{x}_{\xi})^{\top} Q_w(\mathbf{R}\mathbf{w}) - \mathbf{y}_{\xi})^2 \right], \quad (10)$$

Convergence of RoSTE [Theorem 4.3]

Under mild conditions, the weight-activation quantization-aware training of a quantized linear model on least-square loss $\hat{\mathcal{L}}$ converges to

$$\begin{aligned} \mathbb{E}[\hat{\mathcal{L}}(\mathbf{w}^{t+1}, \mathbf{R})] &\leq (1 - \mu)^{t+1} \hat{\mathcal{L}}(\mathbf{w}^0, \mathbf{R}) + (6 + 2\mu^{-1}) \sum_{s=0}^{t+1} (1 - \mu)^{t-s} \|\mathbf{e}(\mathbf{w}^s)\|_{\mathbf{G}}^2 \\ &= \mathcal{O}(\mathbb{E}[\|Q_w(\mathbf{R}\mathbf{w}^t) - \mathbf{R}\mathbf{w}^t\|^2]) \quad \text{when } t \rightarrow \infty \end{aligned} \quad (11)$$

for $0 < \mu < 1$ and $\mathbf{e}(\mathbf{x}) := Q_w(\mathbf{x}) - \mathbf{x}$, i.e., proportional to **weight quantization error** of the converged solution.

Experiments

Table 2: Exp. 1. Accuracies of the 4-bit quantized **Pythia 6.9B** and **Qwen2.5 7B** models fine-tuned using the **Reddit TL;DR** dataset. FP16 and BF16 refer to using 16-bit half-precision floating points and 16-bit brain floating points formats, respectively, and W4A4KV4 refers to using 4-bit quantizations on weights, activation, and KV cache.

Bit-width	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-LSum	ROUGE (Avg.)
Pythia-6.9B						
FP16	Base	28.81	9.45	22.29	22.91	20.87
	SFT	33.69	12.60	26.27	26.31	24.72
W4A4KV4	RTN	7.42	0.06	6.53	6.56	5.14
	GPTQ	8.16	0.08	7.06	7.60	5.73
	QuaRot	11.70	0.23	8.52	9.39	7.46
	SpinQuant	8.61	0.10	8.10	8.07	6.22
	STE	28.91	9.07	22.30	22.33	20.65
	RoSTE	32.60	11.54	25.25	25.25	23.66

Experiments

Bit-width	Method	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-LSum	ROUGE (Avg.)
Qwen2.5-7B						
BF16	Base	32.72	11.82	25.18	25.42	23.79
	SFT	34.75	13.59	27.56	27.58	25.87
W4A4KV4	RTN	1.07	0.00	1.01	1.01	0.77
	GPTQ	0.72	0.00	0.69	0.69	0.53
	QuaRot	7.21	0.10	5.93	5.93	4.79
	SpinQuant	6.87	0.29	5.97	6.12	4.81
	STE	30.86	10.16	23.73	23.73	22.12
	RoSTE	34.01	12.89	26.74	26.74	25.10

Experiments

Table 3: Exp. 2. Accuracies of the 4-bit quantized **Llama 3.1 8B model** fine-tuned on the **Tulu 3** SFT mixture dataset. BF16 refers to using 16-bit brain floating points format, and W4A4KV4 refers to using 4-bit quantizations on weights, activation, and KV cache.

Bit-width	Method	TruthfulQA	MMLU-Pro	BigBenchHard	AGIEval	GSM8K	Math	Avg.
BF16	Base	28.51	19.57	62.26	30.16	56.86	18.20	35.92
	SFT	31.82	33.07	65.67	34.86	64.89	22.66	42.16
W4A4KV4	RTN	23.01	0	0	17.03	1.03	0	6.85
	GPTQ	25.34	0.02	2.55	16.48	2.05	0	7.74
	QuaRot	27.66	21.53	47.69	29.05	37.91	6.90	28.46
	SpinQuant	26.19	21.58	49.56	28.50	38.36	10.56	29.13
	STE	26.68	9.13	24.58	17.63	22.82	1.90	17.14
	RoSTE	26.44	25.12	52.00	30.11	44.50	11.94	31.69

Experiments

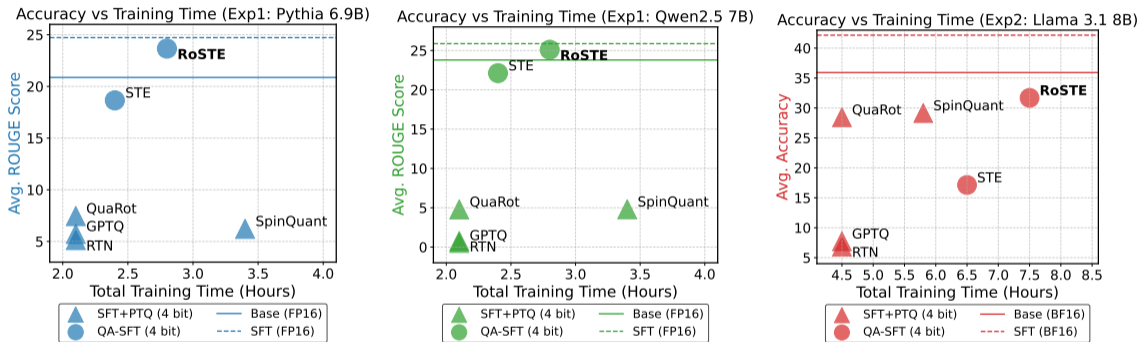






Figure 3: RoSTE surpasses the performance of SOTA quantization methods on fine-tuning benchmark. Horizontal axis represents the total amount of hours needed to fine-tune pre-trained LLMs on a server of $8 \times$ A100 NVIDIA GPUs.

Conclusion

- We proposed the RoSTE algorithm for QA-SFT with an adaptive rotation strategy.
- Besides achieving state-of-the-art performance, we also provide theoretical insights to justify the practical efficacy of RoSTE.
- To the best of our knowledge, this is the first algorithm that leverages adaptive rotation and the fine-tuning objective to produce an accurate quantized model.

Reference I

-  Ashkboos, Saleh, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Martin Jaggi, Dan Alistarh, Torsten Hoefer, and James Hensman (2024). “Quarot: Outlier-free 4-bit inference in rotated llms”. In: *arXiv preprint arXiv:2404.00456*.
-  Courbariaux, Matthieu, Yoshua Bengio, and Jean-Pierre David (2015). “Binaryconnect: Training deep neural networks with binary weights during propagations”. In: *Advances in neural information processing systems* 28.
-  Liu, Zechun, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra (2023). “Llm-qat: Data-free quantization aware training for large language models”. In: *arXiv preprint arXiv:2305.17888*.
-  Tseng, Albert, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa (2024). “Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks”. In: *arXiv preprint arXiv:2402.04396*.