

Late Breaking Results: Quamba-SE: Soft-edge Quantizer for Activations in State Space Models

Yizhi Chen {yizhic@kth.se} and Ahmed Hemani {hemani@kth.se}

Department of Electronics and Embedded Systems, KTH Royal Institute of Technology, Stockholm, Sweden

Abstract—We propose Quamba-SE, a soft-edge quantizer for State Space Model (SSM) activation quantization. Unlike existing methods using standard INT8 operations, Quamba-SE employs three adaptive scales: high-precision for small values, standard scale for normal values, and low-precision for outliers. This preserves outlier information instead of hard clipping, while maintaining precision for other values. We evaluate on Mamba-130M across 6 zero-shot benchmarks. Results show that Quamba-SE consistently outperforms Quamba, achieving up to +2.68% on individual benchmarks and up to +0.83% improvement in the average accuracy of 6 datasets.

Index Terms—Quantization, State Space Models, Quamba

I. INTRODUCTION

Large Language Models (LLMs) are rapidly evolving, dominated by Transformer architectures like GPT. Recently, State Space Models (SSMs), such as Mamba [1], have emerged as significant alternatives. Chiang *et al.* [2] show that Mamba series outperform Pythia [3], the same-sized Transformers.

Quantization is crucial for enhancing speed and reducing storage. For LLMs, Post-Training Quantization (PTQ) [4] is widely used since QAT (Quantization-Aware-Training) [5] requires significant computational training costs. However, related works [2], [4], [6] reveal unique challenges in post-training quantization: SSM activations contain significant outliers. Although outliers constitute a small percentage, including them damages quantization precision for normal values.

Existing methods include Hadamard Transform [2], [6], [7], percentile clipping [2], and group-wise scaling [6]. However, these are CUDA-dependent and lack hardware-level optimization. These methods utilize standard INT8 operations, which have only one scale for all data and hard clip outliers.

This is also a very new question, Quamba [2] and Quamba2 [6] are published in conferences in April 2025 and July 2025, respectively. We propose Quamba-SE, a soft-edge quantizer that operates at the hardware level, providing soft edges for values instead of hard clipping. While evaluated on Quamba, the quantizer design applies to any State Space Models.

II. BACKGROUND AND RELATED WORK

A. Background

We illustrate Quamba and Quamba-SE’s dataflow in Fig. 1. Some components can be easily quantized, such as the INT8 linear projection. The SSM input and output are key challenges for quantization. For the output Y, Hadamard transforms are employed to smooth the outlier distribution.

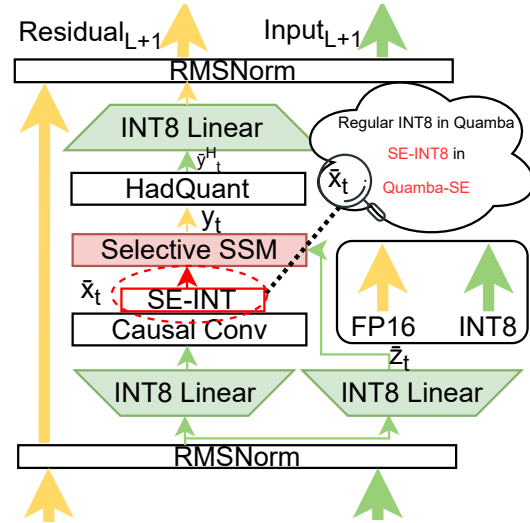


Fig. 1. Quamba-SE: dataflow and data precision

B. Related work

We focus on the SSM input X_t (highlighted in Fig. 1). Mamba-PTQ [4] reports that the challenge of quantizing SSMs stems from activation outliers. Quamba [2] uses calibration, including percentile clipping, to exclude outliers. The stored scales are employed for regular INT8 quantization during inference, achieving higher performance than Mamba-PTQ. Quamba2 [6] extends Quamba with group-wise scales, improving accuracy in Mamba2 [9] models but showing minor improvement in Mamba1 models. Furthermore, Quamba and Quamba2, constrained by CUDA and PyTorch frameworks, focus on finding better scales for standard INT8 operations.

III. OUR METHODS

Same as Quamba and Quamba2, we calibrate and save the scale offline. The difference is during inference as highlighted in red in Fig. 1: Quamba uses a single scale and clips outliers, while we apply three scales adaptively during inference.

We introduce two extra scales in Fig. 2, in addition to using the same stored scale for normal values: a high-precision scale (scale/4) for small values, a standard scale for medium values, and a low-precision scale (scale \times 4) for outliers. During inference, values are first classified: normal values use standard INT8 with the soft-edge (SE) identifier disabled. For values smaller than threshold L or larger than threshold H , the SE identifier is enabled. To avoid extra storage, we use the second bit of INT8 data to distinguish small from large values, leaving

TABLE I
ZERO-SHOT ACCURACY (%) COMPARISON ON MAMBA-130M. BEST IN **BOLD**, SECOND BEST UNDERLINED AMONG QUANTIZED MODELS. [†]AVG. OVER SELECTED BENCHMARKS: LAMBADA, ARC-C, WINOGRANDE.

| Setting | Method | LAMBADA | HellaSwag | PIQA | ARC-E | ARC-C | WinoGrande | Avg. | Δ | Avg. [†] | Δ^{\dagger} |
|---------------------|-----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|----------|-------------------|--------------------|
| Reported [2] | FP16 | 44.20 | 35.30 | 64.50 | 48.00 | 24.30 | 51.90 | 44.70 | – | 40.13 | – |
| | Quamba | 40.60 | 35.00 | 63.00 | 46.50 | 23.00 | 53.10 | 43.50 | – | 38.90 | – |
| FP16 | FP16 | 44.24 | 35.22 | 64.57 | 48.05 | 24.35 | 52.00 | 44.74 | – | 40.20 | – |
| Calib. (99.99%) | Quamba | 42.44 | <u>35.30</u> | <u>63.04</u> | <u>45.92</u> | 23.98 | <u>52.47</u> | <u>43.86</u> | – | 39.63 | – |
| | Quamba-SE | <u>43.42</u> | 34.90 | 62.84 | 46.80 | <u>25.05</u> | 51.51 | 44.08 | +0.22 | <u>39.99</u> | +0.36 |
| Calib. (99.999%) | Quamba | 42.23 | 34.69 | 62.16 | 42.23 | 24.18 | 52.45 | 42.99 | – | 39.62 | – |
| | Quamba-SE | 44.56 | 35.03 | 63.06 | 42.44 | 24.37 | 52.03 | 43.58 | +0.59 | 40.32 | +0.70 |
| Official | Quamba | 39.06 | 35.17 | 61.94 | 45.13 | 24.90 | 50.73 | 42.82 | – | 38.23 | – |
| Weights [8] | Quamba-SE | 40.40 | 35.54 | 62.08 | 45.13 | 25.32 | 53.41 | 43.65 | +0.83 | 39.71 | +1.48 |

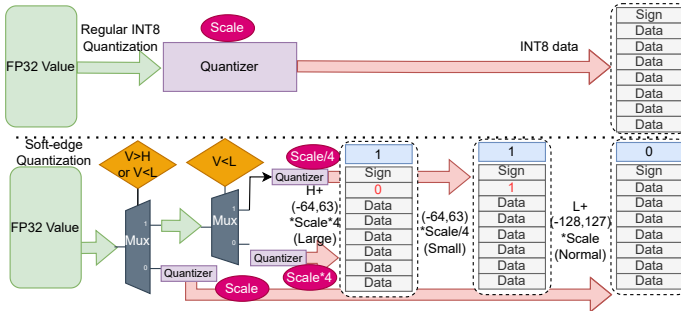


Fig. 2. Hardware architecture of Quamba-SE

6 bits for special-range data. Results show that 6-bit precision for special regions is sufficient. Our solution provides superior dynamic range: high precision for small values, same precision as Quamba without any loss for medium values, and low precision (instead of clipping) for outliers.

In conclusion, while Quamba provides an effective scale, it employs standard INT8 quantization. We adopt Quamba’s effective scale and design a specialized hardware quantizer to provide soft edges for different value ranges.

IV. EXPERIMENTS

We conduct our experiments on the open-source Quamba codebase [10] using an RTX 5090 with CUDA 12.8. As the framework does not support customized INT8, we limit FP32 values to be the data Quamba-SE can represent, to simulate our customized Soft-edge quantizer hardware design. We evaluate on the smallest Mamba model (130M), as quantization is widely used for edge deployment where smaller models are preferred. We test with both official Quamba weights and our generated weights using different percentile calibrations. Following Quamba, we report average accuracy of 5 runs on each dataset, with a total of 6 datasets evaluated.

We present our results compared to Quamba in **Tab. I**, while the reported results [2] serve only as a reference. On LAMBADA, Quamba-SE achieves up to +2.33% improvement (Calib. 99.999%). The highest gain of a single dataset reaches +2.68% on WinoGrande.

Quamba-SE consistently outperforms Quamba across all calibration settings, with +0.22%, +0.59%, +0.83% improvement in average accuracy. Since small models on edge devices

are typically tailored for specific use cases, we further report a subset of benchmarks where the model demonstrates particular strength, showing +0.36%, +0.70%, and +1.48% improvement over Quamba. The largest overall improvement (+1.48% on Avg.[†]) occurs with official pretrained weights. Even when Quamba with 99.99% percentile slightly outperforms reported results in Quamba [2], Quamba-SE still outperforms Quamba.

A. Discussion

Outlier vs Precision Dilemma: Covering outliers increases the step size for quantization; dropping outliers causes information loss. Our soft-edge keeps the precision for most values while retaining outliers instead of hard clipping.

Latency: Our soft-edge quantizer adds latency, but quantization is a minor computation in the whole model, and latency is justified by its accuracy gain. Branch prediction, if applied in future work, can further reduce extra latency overhead.

Quamba2 and Significance: Quamba2 shows minor improvement in Mamba models (Quamba1 57.9%→ Quamba2 58.1% on 1.4B Mamba1). Our Quamba-SE outperforms Quamba1 by 0.83% on 130M Mamba1. In the LLM area, improvements are typically marginal; +0.83% is a meaningful gain given the complexity of multi-dataset evaluation.

V. CONCLUSION

We proposed Quamba-SE, a soft-edge quantizer for Mamba activation quantization that preserves outliers with adaptive scales instead of hard clipping. We evaluate on six datasets with different settings on 130M Mamba model and compare with SOTA work published in 2025. Experiments demonstrate +0.22% to +0.83% accuracy improvement over Quamba.

SSMs’ unique activation features demand specific quantization beyond standard INT8 operations. While existing methods focus on improvements within CUDA constraints, we show that customized hardware quantizers achieve better accuracy. Hardware synthesis and evaluation on Mamba2 with Quamba2 are left for future work. This work represents a promising direction—layer-specific and model-specific quantizer design.

ACKNOWLEDGMENT

The work is partially supported by the EU Horizon projects Cybergy4MIE (101140226) and NeAIxt (101194172).

REFERENCES

- [1] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” in *First conference on language modeling*, 2024.
- [2] H.-Y. Chiang, C.-C. Chang, N. Frumkin, K.-C. Wu, and D. Marculescu, “Quamba: A post-training quantization recipe for selective state space models,” in *The Thirteenth International Conference on Learning Representations*, 2025.
- [3] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O’Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff *et al.*, “Pythia: A suite for analyzing large language models across training and scaling,” in *International Conference on Machine Learning*, 2023, pp. 2397–2430.
- [4] A. Pierro and S. Abreu, “Mamba-PTQ: Outlier channels in recurrent large language models,” in *Workshop on Efficient Systems for Foundation Models II@ ICML*, 2024.
- [5] Z. Liu, B. Oguz, C. Zhao, E. Chang, P. Stock, Y. Mehdad, Y. Shi, R. Krishnamoorthi, and V. Chandra, “Llm-qat: Data-free quantization aware training for large language models,” in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 467–484.
- [6] H.-Y. Chiang, C.-C. Chang, N. Frumkin, K.-C. Wu, M. S. Abdelfattah, and D. Marculescu, “Quamba2: A robust and scalable post-training quantization framework for selective state space models,” in *Forty-second International Conference on Machine Learning*, 2025.
- [7] Z. Xu, Y. Yue, X. Hu, D. Yang, Z. Yuan, Z. Jiang, Z. Chen, S. Zhou *et al.*, “MambaQuant: Quantizing the mamba family with variance aligned rotation methods,” in *The Thirteenth International Conference on Learning Representations*, 2025.
- [8] H.-Y. Chiang, C.-C. Chang, N. Frumkin, K.-C. Wu, M. S. Abdelfattah, and D. Marculescu, “Quamba-130m-w8a8,” 2025, Hugging Face Model Hub: <https://huggingface.co/ut-enyac/quamba-130m-w8a8>.
- [9] T. Dao and A. Gu, “Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality,” in *Forty-first International Conference on Machine Learning*, 2024.
- [10] H.-Y. Chiang, C.-C. Chang, N. Frumkin, K.-C. Wu, and D. Marculescu, “Quamba official github codes repository,” 2025, <https://github.com/enyac-group/Quamba>.