Late Breaking Results: AFS: Improving Accuracy of Quantized Mamba via Aggressive Forgetting Strategy

Zhouquan Liu*†, Libo Huang*†✉, Ling Yang*†, Gang Chen‡, Wei Liu*†✉, Mingche Lai*†, Yongwen Wang*†
*College of Computer Science and Technology, National University of Defense Technology, Changsha, China
†Key Laboratory of Advanced Microprocessor Chips and Systems, Changsha, China
‡School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China

***Abstract*—Mamba overcomes the quadratic complexity problem inherent in Transformer models while maintaining comparable contextual modeling capabilities. However, Mamba-based foundation models encounter challenges in achieving efficient inference on resource-constrained devices, primarily due to their considerable size. Model compression techniques, such as linear quantization, offer a viable solution to this problem. Nevertheless, the introduction of significant outliers during Mamba's past-state forgetting process can lead to a notable decrease in accuracy when employing linear quantization. To overcome these challenges, this paper introduces the Aggressive Forgetting Strategy (AFS), an innovative and efficient algorithm designed to mitigate the quantization issues caused by outliers in the state forgetting mechanism. AFS incorporates a computation-free approach for handling outliers, facilitating both efficient and accurate linear quantization for Mamba, which is essential for applications in resource-constrained scenarios. By leveraging the AFS strategy, Mamba can perform more efficient inference, while significantly improving the accuracy by up to 21.5× compared to conventional methods.**

***Index Terms*—Mamba, foundation model, model compression, outlier-aware quantization, approximation**

I. Introduction

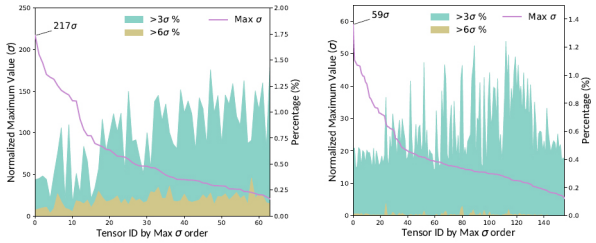
Foundational models (FMs), versatile solutions for diverse problems, are widely recognized as a key component of artificial general intelligence (AGI). Leveraging efficient computation mechanism and powerful contextual modeling performance, Mamba [1] is garnering growing attention as the backbone of FM [2]. However, deploying a Mamba-based FM on resource-constrained devices still faces challenges caused by the explosive growth in the FM's scale. Low-precision linear quantization is one approach to achieve efficient inference of FM. However, due to the significant outliers introduced by the state forgetting operation, it resulted a significant accuracy loss. Previous works [3]–[6] have proposed outlier-aware quantization. However, SmoothQuant [3] and AWQ [4] focus on outliers in the activations, whereas in Mamba, more significant outliers are found in the weights. GOBO [5] and OliVe [6] can handle weight outliers. However, GOBO still requires the high-precision floating-point unit (FPU). While OliVe implements linear quantization, its decoder and the FPU introduced to accommodate outlier calculations also incur additional overhead.

To achieve a balance of accuracy and efficiency for Mamba inference, this paper proposed AFS, AFS utilizes the state forgetting mechanism, avoiding the additional overhead associated with outlier handling. Compared to the vanilla quantization, AFS improved 21.5× accuracy on average under INT8. Compared to SOTA outlier-aware quantization, AFS achieves lower latency while consuming fewer resources.

II. Aggressive Forgetting Strategy

To demonstrate the magnitude of the outliers in the state forgetting parameter $\mathbf{A}$, We normalized the $\mathbf{A}$ in Mamba and all of weights in a Transformer-based FM. Fig. 1 compares the weight distributions, the significant outliers illustrates that linear quantization of parameter $\mathbf{A}$ will results in significant losses.

To implement more efficient linear quantization for the state forgetting operation, we proposed the AFS. In selective scan mechanism (SSM), $\mathbf{A}$ is constrained to the negative value, while $\mathbf{\Delta}$ is constrained to the positive value. $\mathbf{A}$ interacts with $\mathbf{\Delta}$ to derive input-dependent selectivity, i.e., $\exp(\mathbf{\Delta A})$. We observe that the exponential function amplifies differences in the inputs, which implies that the output for outliers of $\mathbf{A}$ is most likely 0, unless $\mathbf{\Delta}$ is very close to 0. By leveraging the sparsity of the output, we can more aggressively pre-filter outliers from $\mathbf{A}$ and quantize only the normal values. Specifically, the AFS can be divided into three steps as follows.



(a) Mamba-2.8b (b) Gemma2-2b

Fig. 1: Weight distribution of foundation models. (a) the distribution of normalized $\mathbf{A}$ in Mamba. (b) the distribution of all weights in Gemma2.

Grouping. In the distribution of $\mathbf{A}$, the majority are negative values close to 0, along with some negative values of very large magnitude. This skewed distribution affects the applicability of using probability density function (PDF) for outlier filtering. Therefore, we first mirror $\mathbf{A}$ along the y-axis, and then apply the interquartile range (IQR) method for grouping. Values below the lower bound $Q1 - 1.5 * IQR$ are considered outliers.

978-3-9826741-0-0/DATE25/© 2025 EDAA

TABLE I: Comparison with vanilla quantization on six datasets.

Dataset	Metric	Mamba-1.4b			Mamba-2.8b		
		FP32	w/o AFS	w AFS	FP32	w/o AFS	w AFS
LAMBADA	PPL ↓	5.04	2E+11	**6.52**	4.23	2E+3	**4.29**
	ACC ↑	64.9	0.0	**59.5**	69.2	0.0	**68.4**
HELLASWAG	ACC ↑	59.1	28.8	**57.4**	66.2	37.1	**65.0**
ARC-E	ACC ↑	65.5	44.2	**64.8**	69.7	63.4	**69.8**
ARC-C	ACC ↑	32.9	24.0	**32.3**	36.3	31.2	**36.8**
WINOGRANDE	ACC ↑	61.4	51.9	**59.7**	63.4	60.5	**63.2**
AVERAGE	ACC ↑	56.8	29.8	**54.7**	61.0	38.5	**60.6**
ACCURACY LOSS ↓			27.0	**2.0**		22.5	**0.3**

Encoding. After grouping, the normal values are linearly mapped to a low-precision integer range $-(2^{num\_bits}-1) \sim 2^{num\_bits}-1$. A specific minimum value -2^{num_bits} is reserved as the outlier identifier. Since $\mathbf{A}$ is constrained to negative values, we use asymmetric quantization to enhance accuracy.

Filtering. The grouping and encoding process can be performed at the offline phase to enhance inference efficiency. At runtime, AFS checks the encoding of $\mathbf{A}$. When AFS detects an outlier identifier, it sets the *flag* to 1; otherwise, the *flag* is set to 0. The exponent unit ignores values corresponding to a *flag* of 1 and directly outputs 0 for them. For values with a *flag* of 0, their product is considered valid.

III. EVALUATION

A. Experiment Setup

Models and Datasets. To evaluate the accuracy of AFS, we conducted ablation studies with Mamba-1.4b and Mamba-2.8b using LAMBADA, HellaSwag, Arc-E, Arc-C and WinoGrande datasets. We evaluated the FP32 models[1] and the INT8 models quantized with and without AFS using *lm-eval*[2].

Hardware Efficiency. We synthesize AFS and the SOTA outlier-aware method [6] with FPGA. All hardware designs were implemented using SpinalHDL[3] and synthesized at 200 MHz on XCVU19P via Vivado 2023.1.

B. Accuracy Loss

Tab. I compares the accuracy loss of state forgetting quantization between with and without AFS. Due to the impact of outliers, linear integer quantization can lead to significant accuracy losses. In contrast, AFS achieves a balance of accuracy and efficiency by leveraging the computational mechanism. Under six datasets, AFS improves accuracy by 21.5× compared to vanilla quantization.

C. Compared to SOTA methods

Tab. II and Tab. III compares AFS with state-of-the-art (SOTA) weight outlier-aware quantization. Compared to GOBO [5], AFS enables efficient linear quantization. Compared to OliVe [6], AFS offers higher hardware efficiency. This benefit stems from the avoidance of floating-point encoding and decoding for outliers, with a sacrifice in negligible input selectivity.

[1]https://huggingface.co/state-spaces/

[2]https://github.com/EleutherAI/lm-evaluation-harness

[3]https://github.com/SpinalHDL/SpinalHDL

TABLE II: Comparison with previous methods.

	GOBO [5]	OliVe [6]	Ours
TYPE	non-linear	linear	linear
STORAGE BIT WIDTH	3	4	8
ALIGNED MEMORY	unaligned	aligned	aligned
OUTLIER HANDLING	FP32 Mul	FP8 Mul	INT8 Mul
HARDWARE OVERHEAD	high	medium	low

TABLE III: Comparison of hardware efficiency with OliVe. The results demonstrate the overhead of a multiplication unit runs at 200 MHz.

		OliVe [6]	Ours
RESOURCE CONSUMPTION	LUT	118	65
	FF	65	18
LATENCY(CYCLE)		4	1

IV. CONCLUSION

While Mamba alleviates the Transformer's quadratic complexity via SSM, deploying it on resource-constrained devices is still not trivial. The primary challenge stems from the outliers in the forgetting gate. We propose AFS to enhance low-precision integer quantization accuracy, enabling efficient linear quantization for inference. AFS leverages the state-forgetting mechanism for efficient outlier handling. Compared to standard quantization, AFS reduces accuracy loss by up to 21.5×. Furthermore, AFS exhibits superior hardware efficiency compared to previous outlier-aware quantization methods.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Grant Nos.62102433 and 62272475), in part by the Provincial Natural Science Foundation of Hunan (Grant2022JJ10064), , and part by the Key Laboratory of Advanced Microprocessor Chips and Systems.

REFERENCES

[1] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," May 2024, arXiv:2312.00752.

[2] L. Ren, Y. Liu, Y. Lu, Y. Shen, C. Liang, and W. Chen, "Samba: Simple Hybrid State Space Models for Efficient Unlimited Context Language Modeling," Jun. 2024.

[3] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models," in *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023.

[4] J. Lin *et al.*, "AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration," *Proceedings of Machine Learning and Systems*, vol. 6, pp. 87–100, 2024.

[5] A. H. Zadeh, I. Edo, O. M. Awad, and A. Moshovos, "GOBO: Quantizing Attention-Based NLP Models for Low Latency and Energy Efficient Inference," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Oct. 2020, pp. 811–824.

[6] C. Guo *et al.*, "OliVe: Accelerating Large Language Models via Hardware-friendly Outlier-Victim Pair Quantization," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ser. ISCA '23. New York, NY, USA: Association for Computing Machinery, Jun. 2023, pp. 1–15.