OuroMamba: A Data-Free Quantization Framework for Vision Mamba
Supplementary Material

A. Extended Related Works

Existing PTQ techniques for ViTs [6, 10, 12, 14, 16, 17, 21, 22] address long-tailed distributions and static activation outliers to enhance quantization accuracy. For instance, DopQ-ViT [21] and ADFQ-ViT [6] mitigate outliers by optimizing per-channel and per-patch scale factors, respectively. FQ-ViT [12] introduces Power-of-Two Factor (PTF) for inter-channel LayerNorm variation and Log-Int-Softmax (LIS) for 4-bit attention map quantization. Among the early PTQ methods for Mamba models, Mamba-PTQ [15] and Quamba [3] identified activation outliers as a key challenge but were tailored for language tasks. More recently, VMM PTQ techniques [4, 9] highlighted the highly dynamic activation distributions and inter-channel variations across time-steps. PTQ4VM [4] adapts SmoothQuant [20] to migrate activation outliers into weights using a migration factor. However, as noted in [7], this increases weight complexity, making both weights and activations more sensitive to dynamic variations, rendering it ineffective for ultra-low precision (< 4 bits) quantization. Additionally, PTQ4VM does not quantize SSM operators, limiting its scope to linear layer weights and output activations. QMamba [9] addresses the dynamic inter-time-step variations in VMMs' hidden states by introducing fine-grained temporal grouped quantization, quantizing both weights and activations. Similarly, kSQ-VMM [18] applies similarity-based k-scaled channel-wise and token-wise quantization to handle dynamic activation distributions. However, existing VMM PTQ methods rely on static scale factors [4, 9] or fixed temporal groupings [9], leading to accuracy degradation at ultra-low bit precisions due to their inability to dynamically manage outlier channels.

B. OuroMamba DFQ Algorithm

B.1. OuroMamba-Gen

In Algorithm 1, we detail the OuroMamba-Gen pipeline.

B.2. OuroMamba-Quant

In Algorithm 2, we detail the OuroMamba-Quant pipeline.

Algorithm 1: OuroMamba-Gen

Input: A pre-trained FP VMM model P with L layers, Gaussian noise batch $X_{\mathcal{B}}$, task-specific targets $T_{G_{\mathcal{B}}}$, neighborhood size $\mathcal{N}$, iterations G.
Output: A set of generated synthetic samples $X^*_{\mathcal{B}}$.

for $g = 1, 2, \dots, G$ **do**
  Input $X_{\mathcal{B}}$ into P ;
  for $l = 1, 2, \dots, L$ **do**
    Capture per-time-step original $h^l(t)$, $\Delta^l(t)$;
    Set $w^l_t = \mathtt{mean}_E(\Delta^l(t))$;
    Compute $h^l_p(t)$;
    Extract implicit attention;
    Compute $\mathcal{L}^C_l = \sum_t \mathcal{L}^C_{l,t}$;
  end
  Compute $\mathcal{L}^C = \sum_l \mathcal{L}^C_l$;
  Compute output loss $\mathcal{L}^O$;
  Compute $L^{gen} = \mathcal{L}^C + \mathcal{L}^O$;
  Update $X^*_{\mathcal{B}}$ via backpropagation of L^{gen};
end

C. Additional Quantization Results

In Table 1 we provide additional quantization results of Vim-T [24], VMamba-T [13] and the hybrid model MambaVision-T [5] for image classification.

D. Additional Ablations

W8A8 Quantization. In Table 2, we compare PTQ4VM with OuroMamba, following the experimental setup outlined in Sec. 6.1. It is important to note that here, $b^O_a = 16$.
Real v/s Synthetic Samples. In Table 4, we compare the accuracy with real and OuroMamba-Gen synthetic calibration samples on Vim-S, using 128 images for both. Notably, the synthetic samples closely match the accuracy achieved via real images.
Outlier Detection. Table 5 presents the classification accuracy of Vim-B under different outlier detection mechanisms, highlighting their impact on quantized model per-

Algorithm 2: OuroMamba-`Quant`

Input : Activation $X(t) \in \mathbb{R}^{N \times E}$, Static scale $S^I(t)$, Threshold θ, Refresh rate $n_{\texttt{refresh}}$, Outlier list $O_{\texttt{list}}$, Inlier and outlier bit-precision b_a^I, b_a^O

Output: Quantized activation $X_q(t)$, Updated outlier list $O_{\texttt{list}}$

if t % $n_{refresh} == 0$ **then**
    $O_{\texttt{list}} = \{\phi\}$
end
$S^D(t) = \texttt{ComputeScale}(X(t)[:, c] \,\forall\, c \notin O_{\text{list}})$
if $S^D(t) > S^I(t)$ **then**
    for *each channel c in* $X(t)$ ***not in*** O_{list} **do**
        if $max(|X(t)[:, c]|) \geq \theta$ **then**
            $O_{\texttt{list}} = O_{\texttt{list}} \cup \{c\}$
        end
    end
end
$I(t), O(t) = \texttt{Separate}(X(t), O_{\texttt{list}})$
$I_q(t) = \texttt{InlierQuant}(I(t), S^I(t), b_a^I)$
$O_q(t) = \texttt{OutlierQuant}(O(t), b_a^O)$
$X_q(t) = \texttt{Merge}(I_q(t), O_q(t))$
return $X_q(t), O_{\texttt{list}}$

Table 1. Quantization accuracy comparison of SoTA techniques on ImageNet classification. 'R', 'S' signifies real and synthetic calibration data.

	Method	Data	#Images	W/A	Top-1	W/A	Top-1	W/A	Top-1
Vim-T [24]	Baseline	-	-	32/32	76.10	32/32	76.10	32/32	76.10
	PTQ4VM [4]	R	256	4/8	74.15	6/6	73.94	4/4	56.29
	QMamba [9]	R	1024	4/8	70.13	6/6	57.95	4/4	53.41
	OuroMamba (Ours)	S	128	4/8	**74.98**	6/6	**74.84**	4/4	**63.49**
VMamba-T [13]	Baseline	-	-	32/32	82.60	32/32	82.60	32/32	82.60
	PTQ4VM [4]	R	256	4/8	77.02	6/6	75.67	4/4	72.67
	QMamba [9]	R	1024	4/8	76.51	6/6	**80.49**	4/4	51.48
	OuroMamba (Ours)	S	128	4/8	**81.73**	6/6	80.15	4/4	**77.56**
Hybrid Model MambaVision-T [5]	Baseline	-	-	32/32	82.30	32/32	82.30	32/32	82.30
	PTQ4VM [4]	R	256	4/8	72.13	6/6	69.39	4/4	67.67
	QMamba [9]	R	1024	4/8	71.93	6/6	68.17	4/4	65.33
	OuroMamba (Ours)	S	128	4/8	**80.57**	6/6	**79.05**	4/4	**74.92**

Table 2. W8A8 quantization accuracy on ImageNet.

Model	Method	Data	Top-1
Vim-S	FP Baseline	-	81.60
	PTQ4VM	R	81.23
	OuroMamba	S	**81.42**
Vim-B	FP Baseline	-	81.90
	PTQ4VM	R	**80.30**
	OuroMamba	S	80.18

Table 3. L40S GPU speedup results (Batch Size = 32).

Model	Method	Speedup
Vim-S	PTQ4VM	$1.23\times$
	OuroMamba	$\mathbf{1.40}\times$
Vim-B	PTQ4VM	$1.29\times$
	OuroMamba	$\mathbf{2.06}\times$
VMamba-B	PTQ4VM	$1.93\times$
	OuroMamba	$\mathbf{2.37}\times$
MambaVision-T	PTQ4VM	$1.39\times$
	OuroMamba	$\mathbf{1.70}\times$

formance. In Table 5, 'None' indicates no outlier detection, 'Static' indicates a statically (offline) identified list of 64 outlier channels, and 'Dynamic' corresponds to our proposed scheme. Evidently, using 'Dynamic' outlier detection offers the best quantized model accuracy with **up to** $\mathbf{20.61}\%$ increase in accuracy over 'Static'.

Table 4. Ablation of real v/s synthetic samples.

Model, W/A	Data	Top-1
Vim-S 4/8	R	**79.92**
	S	79.81
Vim-S 4/4	R	**75.93**
	S	**75.93**

Table 5. Outlier detection ablation

Model, W/A	Outlier Det.	Top-1
Vim-B 4/8	None	74.28
	Static	76.87
	Dynamic (Ours)	**80.17**
Vim-B 4/4	None	0.10
	Static	56.73
	Dynamic (Ours)	**77.34**

E. GEMM Implementation Details

We first describe how the GEMM operation can be decomposed into separate computations for outliers and inliers. Consider an output element $Y[i, j]$ computed as

$$\begin{aligned} Y[i,j] &= \sum_{k=0}^{K-1} A[i,k]\, W[k,j], \\ &= \sum_{k \in \mathcal{I}} A^I[i,k]\, W[k,j] + \sum_{k \in \mathcal{O}} A^O[i,k]\, W[k,j], \end{aligned}$$

where the inlier activations A^I have the outlier positions zeroed out, and the outlier activations A^O have the inlier positions zeroed. This decomposition guarantees that the sum of the two partial GEMM results yields the same $Y[i, j]$ as the original full GEMM.

We now introduce our GEMM pipeline, which consists of the following five steps:

1. **Outlier Extraction**
Outlier values in the input activation are identified, and their corresponding positions are zeroed out. The outlier columns are then compacted into a small INT8 outlier buffer.
2. **Inlier Extraction**
With the outlier positions already zeroed out, the inlier values are extracted and packed into INT4 buffers, storing two values per byte.
3. **INT4 GEMM**
An INT4 GEMM is performed on the inlier data. During the CUTLASS epilogue, the results are immediately dequantized by multiplying with the activation and weight scales. This fusion is enabled by the use of per-tensor quantization for inliers, which offers greater efficiency compared to the per-token inlier quantization employed by PTQ4VM.
4. **INT8 GEMM**
A mixed-input INT4-INT8 GEMM is executed between the inlier and the compacted outlier matrices, utilizing INT8 tensor cores.
5. **Outlier Dequantization and Combination**
Finally, the dequantization of outliers and the combination of the two GEMM results are fused into a single kernel. This kernel is memory-bound because it writes the final result matrix.

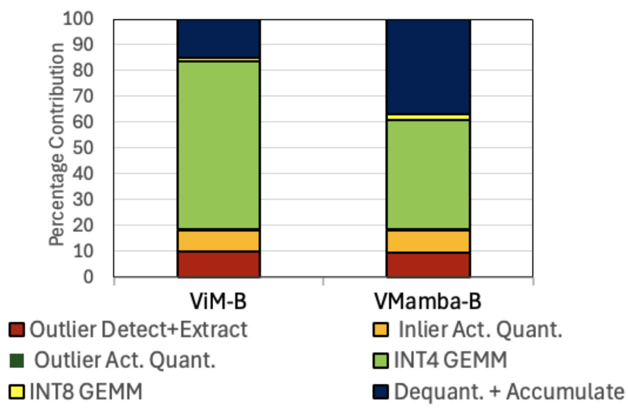

Figure 1. Kernel breakdown of OuroMamba-`Quant`

Table 6. Memory compression comparison over PTQ4VM [4].

	Vim-S [24]		Vim-B [24]	
Method	W/A	Mem. Comp.	W/A	Mem. Comp.
Baseline	16/16	1.00	16/16	1.00
PTQ4VM [4]	4/4	1.81	4/4	2.02
OuroMamba (Ours)	4/4	**3.63**	4/4	**3.80**

F. Speed Breakdown Results

As shown in Fig. 1, outlier extraction incurs minimal overhead. Specifically, we partition activation by channels, so outlier channel indices and scaling factors are calculated and recorded in parallel. Additionally, our compact extraction approach restricts the INT8 GEMM operation to a small subset of outlier channels, limiting its runtime contribution to less than 5%. As expected, the INT4 GEMM remains the dominant component. The primary performance bottleneck is the dequantization and combination step. This step writes to the entire output matrix, making it inherently memory-bound and therefore more expensive. Notably, the dequantization overhead is higher in Vmamba-B because it has a higher outlier rate (4.3%) compared to Vim-B (1.3%). Nonetheless, even in scenarios with higher outlier densities, our overall pipeline remains efficient due to the minimal costs associated with both outlier extraction and the outlier GEMM computations.

F.1. Additional GPU Speedup Results

In the main draft, Fig. 8 and Sec. 6.7 discusses speedups on an A100 GPU for classification, generation tasks. Additionally, in Table 3 we report speedups for four models on the classification task on a workstation-grade L40S GPU.

G. Memory Compression Results

As shown in Table 6, we compare the memory compression factor of OuroMamba with PTQ4VM at W4A4, using the FP16 model as the baseline, on the Vim-S and Vim-B models [24]. The results show that OuroMamba consistently achieves a high memory compression factor of up to 3.80× compared to the baseline FP16 model, while PTQ4VM achieves a memory compression factor of only 2.02×, as it quantizes only the Linear layers of VMMs.

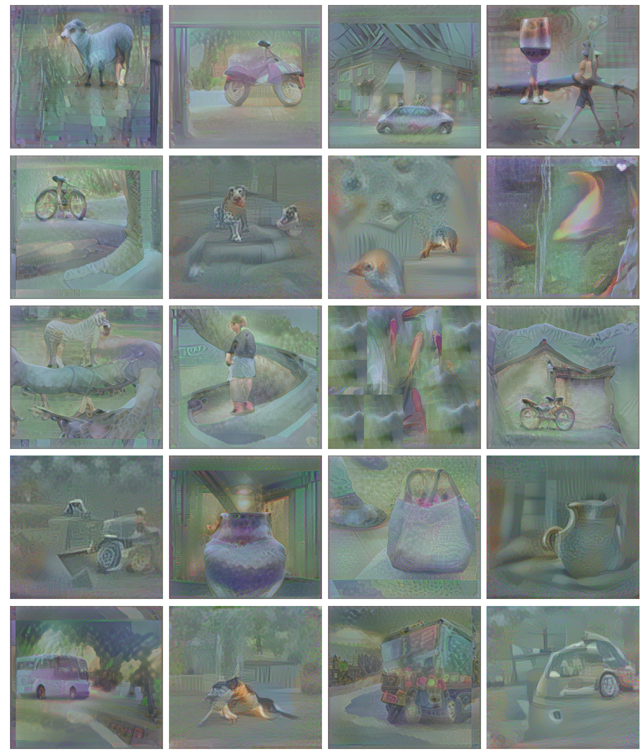

Figure 2. Generated synthetic data samples.

H. Extension of OuroMamba-`Quant` to Transformer based models

We extend OuroMamba-`Gen` to Transformer-based models and layers by mapping the time-step dimension to the token dimension. Outlier channels are identified per token, with $O_{\texttt{list}}$ propagated across tokens.

I. Text-to-Image Generation Results

Implementation. We applied OuroMamba-`Quant` to PixArt-Σ [1] with 20-iteration setting. Following ViDiT-Q [23], we quantize linear layers for query, key, and value projections and the second projection layer of feed-forward network to W4A4. Meanwhile, the first projection layer of feed-forward network and the output projection in self-attention are quantized in 8-bit for better numerical stability, while outliers bits are fixed at 8-bit and $n_{\texttt{refresh}}$ is set to 10. For calibration, we follow Q-Diffusion [8] and randomly sample text prompts from MS-COCO dataset [11] to obtain outlier threshold and inlier scale factors.

Results. In Fig. 3, we visualize the generated images of W4A8, W4A4 Ouromamba-`Quant` quantized PixArt-Σ compared to W4A8 Q-DiT [2] and W4A8 PTQ4DiT [19].

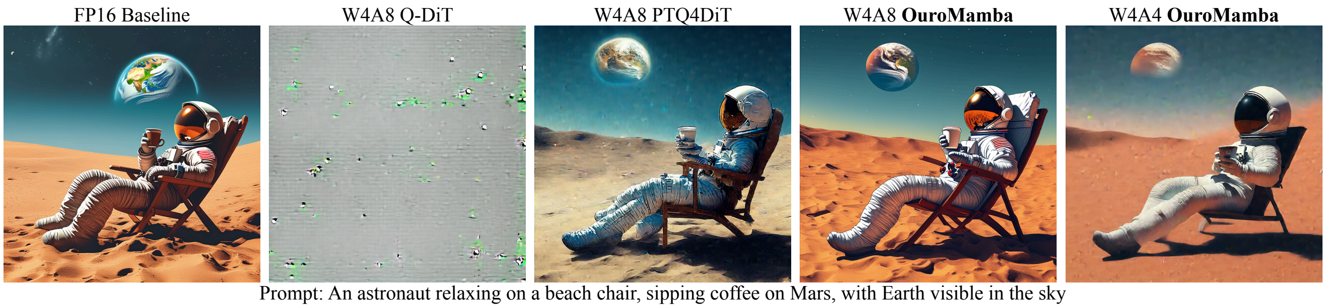

Figure 3. Quantization performance comparison for text-to-image generation task.

J. Additional Synthetic data samples

In Fig. 2, we additionally visualize synthetic samples generated by OuroMamba-`Gen` for image classification, object detection and segmentation tasks for Vim-B model [24].

References

[1] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024. 3

[2] Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-dit: Accurate post-training quantization for diffusion transformers, 2024. 3

[3] Hung-Yueh Chiang, Chi-Chih Chang, Natalia Frumkin, Kai-Chiang Wu, and Diana Marculescu. Quamba: A post-training quantization recipe for selective state space models. *arXiv preprint arXiv:2410.13229*, 2024. 1

[4] Younghyun Cho, Changhun Lee, Seonggon Kim, and Eunhyeok Park. PTQ4VM: Post-training quantization for visual mamba. *Winter Conference on Application of Computer Vision*, 2025. 1, 2, 3

[5] Ali Hatamizadeh and Jan Kautz. Mambavision: A hybrid mamba-transformer vision backbone. *In Proceedings of the Computer Vision and Pattern Recognition*, 2025. 1, 2

[6] Yanfeng Jiang, Ning Sun, Xueshuo Xie, Fei Yang, and Tao Li. Adfq-vit: Activation-distribution-friendly post-training quantization for vision transformers. *Neural Networks*, page 107289, 2025. 1

[7] Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models, 2025. 1

[8] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models, 2023. 3

[9] Yinglong Li, Xiaoyu Liu, Jiacheng Li, Ruikang Xu, Yinda Chen, and Zhiwei Xiong. Qmamba: Post-training quantization for vision state space models. *arXiv preprint arXiv:2501.13624*, 2025. 1, 2

[10] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17227–17236, 2023. 1

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 3

[12] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. *arXiv preprint arXiv:2111.13824*, 2021. 1

[13] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2025. 1, 2

[14] Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. Outlier-aware slicing for post-training quantization in vision transformer. In *Forty-first International Conference on Machine Learning*, 2024. 1

[15] Alessandro Pierro and Steven Abreu. Mamba-ptq: Outlier channels in recurrent large language models. *arXiv preprint arXiv:2407.12397*, 2024. 1

[16] Akshat Ramachandran, Souvik Kundu, and Tushar Krishna. Clamp-ViT: Contrastive data-free learning for adaptive post-training quantization of vits. In *European Conference on Computer Vision*, pages 307–325. Springer, 2024. 1

[17] Akshat Ramachandran, Zishen Wan, Geonhwa Jeong, John Gustafson, and Tushar Krishna. Algorithm-hardware co-design of distribution-aware logarithmic-posit encodings for efficient dnn inference. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pages 1–6, 2024. 1

[18] Bo-Yun Shi, Yi-Cheng Lo, et al. Post-training quantization for vision mamba with k-scaled quantization and reparameterization. *arXiv preprint arXiv:2501.16738*, 2025. 1

[19] Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers, 2024. 3

[20] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and effi-

cient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR, 2023. 1

[21] Lianwei Yang, Haisong Gong, and Qingyi Gu. Dopq-vit: Towards distribution-friendly and outlier-aware post-training quantization for vision transformers. *arXiv preprint arXiv:2408.03291*, 2024. 1

[22] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *European conference on computer vision*, pages 191–207. Springer, 2022. 1

[23] Tianchen Zhao, Tongcheng Fang, Haofeng Huang, Enshu Liu, Rui Wan, Widyadewi Soedarmadji, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. Vidit-q: Efficient and accurate quantization of diffusion transformers for image and video generation, 2025. 3

[24] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model, 2024. 1, 2, 3, 4