Hardware-Friendly Token-Group Activation Quantization for Low-Bit Mamba Super-Resolution

Jinwoo Chung
Kookmin University
Seoul, Republic of Korea
imaboybut@kookmin.ac.kr

Jangho Kim*
Kookmin University
Seoul, Republic of Korea
jangho.kim@kookmin.ac.kr

Abstract

Mamba-based super-resolution models are efficient alternatives to Transformer-based restoration backbones, but their low-bit post-training quantization remains challenging because selective scan, Mamba's input-dependent state-space operator, induces different activation ranges across spatial tokens. Thus, a single layer-wise clipping range poorly matches all tokens. We propose *Token-Group Quantization*, a hardware-friendly activation PTQ framework that statically approximates a hardware-unfriendly post-scan grouping oracle, where tokens with similar preferred clipping ranges are assigned to the same group. To avoid runtime post-scan grouping in static low-bit inference, a lightweight predictor infers the groups from pre-scan features available before the quantized scan path. During calibration, we form pseudo-labels by clustering post-scan activation descriptors that capture similar rounding–clipping behavior. At inference, the frozen predictor maps pre-scan features to groups and selects entries from offline-calibrated group-wise clipping-bound tables, without runtime statistics collection, dynamic clipping updates, or post-scan group assignment. To address low-bit restoration degradation, we further introduce a Frequency-Preserving Refinement (FPR) objective for clipping-bound calibration. FPR aligns Fourier amplitude and phase between full-precision teacher and quantized student outputs to preserve high-frequency structures. Across PTQ protocols, bit-widths, backbones, and tasks, our method consistently outperforms prior Mamba and SR PTQ baselines, transfers to Transformer SR and Vision Mamba classification, composes with rotation-based quantization, and achieves $2.23\times$ speedup on Jetson Orin Nano over FP16. Code is available here.

1 Introduction

Single image super-resolution (SR) reconstructs high-resolution images from low-resolution inputs. Recent SR backbones have progressed from CNNs to Transformers, where long-range dependency modeling improves reconstruction fidelity [29, 48, 28, 4]. More recently, Mamba-based SR models have emerged as efficient alternatives by replacing quadratic self-attention with selective state-space modeling, capturing long-range spatial dependencies with linear complexity [15–17]. Selective scan is the core input-dependent state-space operator in this family, propagating information across flattened spatial tokens; in MambaIRv2, our primary backbone, it is implemented inside Attentive State-Space Modules (ASSMs). While this improves architectural efficiency, edge deployment still requires reducing activation memory traffic and enabling low-bit fixed-point execution, motivating the study of post-training quantization (PTQ) for Mamba-based SR.

Although PTQ compresses pretrained models without updating the backbone weights [36, 27], conventional PTQ severely degrades low-bit Mamba SR. We identify a key cause. Selective scan

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

substantially amplifies token-wise activation heterogeneity. As shown in Sec. 4, in MambaIRv2-Light, 18 out of 20 ASSM blocks show increased per-token range variation after selective scan, with an average $1.80\times$ increase over pre-scan activations. Thus, the bottleneck is not a few rare outliers, but broad token-wise range heterogeneity: a loose range wastes precision on typical tokens, while a tight range clips high-response tokens critical for restoration.

This mismatch suggests that different token groups should be quantized under different clipping ranges. The most direct way to obtain such groups is to inspect post-scan activations at the target quantization sites and assign each token to a group accordingly, since these activations reveal the clipping range preferred at those sites. However, this post-scan group assignment is hardware-unfriendly for static low-bit deployment. It requires runtime statistics collection inside the quantized path and complicates ahead-of-time compilation with low-bit fixed-point kernels. We call this hardware-unfriendly upper-bound grouping rule the *downstream oracle*, a post-scan grouping oracle instantiated by K-means over post-scan activation descriptors that encode rounding–clipping regimes. The challenge is therefore to recover the oracle's grouping behavior under a static deployment constraint, where clipping-bound tables and quantization parameters remain fixed after calibration.

Building on this observation, we propose **Token-Group Quantization**, which distills the downstream oracle's grouping behavior into a static pre-scan prediction problem. This approximation is feasible because selective scan generates token-dependent recurrence parameters from pre-scan features, which therefore already encode signals predictive of downstream activation regimes. During offline calibration, quantization-aware pseudo-labels capture token groups that should share clipping ranges, and a lightweight shared predictor learns to assign tokens to these groups from pre-scan features. At inference, the frozen predictor selects entries from clipping-bound tables indexed by layer, quantization site, and group, avoiding runtime statistics collection, dynamic clipping updates, and post-scan group assignment.

Beyond activation range mismatch, low-bit SR quantization also degrades high-frequency edges and textures, which pixel-space losses can under-supervise. To address this restoration-specific degradation, we introduce a teacher-supervised Frequency-Preserving Refinement (FPR) objective for clipping-bound calibration, which aligns Fourier amplitude and phase between full-precision teacher and quantized student outputs. Our main contributions are summarized as follows:

- We identify selective-scan-induced token-wise activation heterogeneity as a key failure mode of low-bit PTQ for Mamba-based SR, showing through pre- and post-scan analysis that selective scan consistently amplifies token-range variation across most ASSM blocks.
- We propose *Token-Group Quantization*, which frames token-adaptive low-bit quantization as a static approximation to a hardware-unfriendly post-scan grouping oracle. During calibration, we instantiate this oracle by K-means over downstream activation descriptors and train a lightweight pre-scan predictor to recover oracle-aligned groups before the quantized sites.
- We introduce a teacher-supervised Frequency-Preserving Refinement (FPR) objective for low-bit SR PTQ, which refines clipping bounds by aligning Fourier amplitude and phase between full-precision teacher and quantized student outputs. Experiments show consistent gains over prior Mamba and SR PTQ baselines with $2.23\times$ speedup on Jetson Orin Nano.
- The gains hold across MambaIR v1 and v2 at Light and Base scales and across denoising and JPEG artifact removal, transfer to SwinIR-light and to ImageNet on Vision Mamba, and compose with channel-axis transforms such as SpinQuant, as shown in Appendices A and B.

2 Related work

2.1 Image Super-Resolution

Single image super-resolution recovers high-resolution images from low-resolution inputs. Early SR models were dominated by CNNs such as SRCNN [12], VDSR [23], EDSR [29], and RCAN [48], while Transformer-based models such as SwinIR [28] and HAT [4] improve restoration quality by modeling long-range dependencies at higher compute and memory cost. State-space models with selective scan [15] have been introduced to vision [51, 31] as efficient long-range backbones, and MambaIR [16] and MambaIRv2 [17] adapt them to image restoration by combining attention,

state-space blocks, and convolution. Despite their efficiency, edge deployment of Mamba-based SR remains challenging due to activation memory traffic and low-bit quantization sensitivity.

2.2 Post-Training Quantization

Model compression techniques, including pruning [22] and quantization, aim to reduce deployment cost. Quantization is commonly divided into quantization-aware training (QAT) and post-training quantization (PTQ). QAT simulates quantization during training and can achieve high accuracy, but requires costly retraining and careful tuning [13, 25, 50, 9], including recent SSM restoration work such as Q-MambaIR [5]. PTQ instead keeps the backbone weights fixed and calibrates quantization parameters from a small dataset [2, 36, 27, 20], but low-bit PTQ is vulnerable to activation outliers and input-dependent distribution shifts. Recent studies further go beyond simulated (fake) quantization and demonstrate actual low-bit acceleration on hardware [49, 1, 11, 10, 24].

For SR, where activation statistics vary across spatial locations and tokens, prior methods use clipping-bound calibration and outlier-aware designs [39, 30, 40].

Mamba-specific PTQ methods such as PTQ4VM [8], Quamba [6], and MambaQuant [46] target selective state-space architectures, but are mainly validated outside SR restoration and still degrade substantially on Mamba-based SR. Weight-training PTQ methods such as AIQViT [21] and SPR^{2}Q [45] compensate quantization error with LoRA-style adapters [18] fused before deployment. Rotation, affine, and scaling transforms from LLM and diffusion quantization [1, 32, 33, 37, 44, 26] remove channel-wise outliers, and Quamba and MambaQuant apply the same idea to Mamba. All of these assign the quantization range independently of token identity, which is sound when token distributions are homogeneous but is exactly the assumption that selective scan breaks in Sec. 4. Atom [49] acts on the token axis but relies on runtime statistics. Token-Group Quantization instead assigns ranges per token group statically through offline-calibrated clipping-bound tables and a lightweight pre-scan predictor, and it is complementary to channel-axis transforms, as combining it with rotation-based methods such as SpinQuant yields further gains in Appendix B.4.

3 Preliminaries

3.1 Selective Scan in Mamba-based SR

A discretized state-space model updates a hidden state h_t and produces an output y_t as

$$h_t = \bar{A}_t h_{t-1} + \bar{B}_t x_t, \qquad y_t = C_t h_t + D x_t. \tag{1}$$

Unlike linear time-invariant SSMs, Mamba makes recurrence input-dependent by generating token-specific parameters from x_t. Thus, different tokens can induce different recurrence dynamics.

In Mamba-based SR, selective scan is applied to flattened spatial tokens inside restoration blocks, where an image feature map is represented as a sequence of length $L = HW$. Since the recurrence parameters are token-dependent, selective scan can produce different post-scan activation regimes across spatial locations. This property motivates our analysis of token-wise post-scan activation heterogeneity in Section 4.

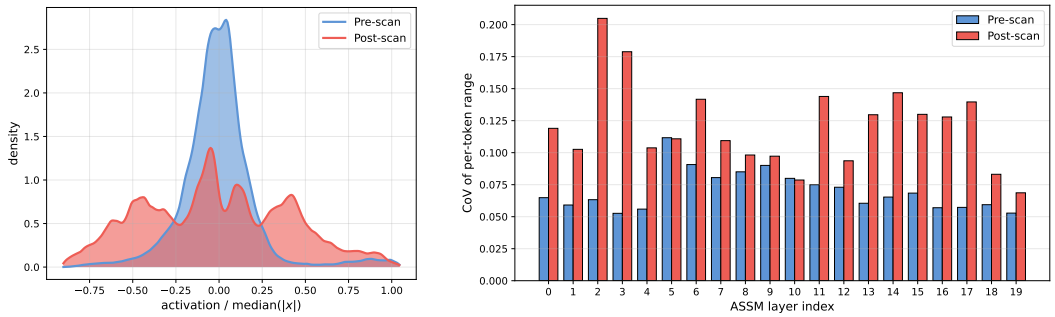
3.2 Uniform Quantization for PTQ

Post-training quantization (PTQ) converts a pretrained full-precision model into a low-bit model by calibrating quantization parameters while keeping the backbone weights fixed. For an activation tensor x, we use a b-bit uniform quantizer with lower and upper clipping bounds l and u:

$$Q(x; l, u, b) = s \cdot \mathrm{clip}\left(\left\lfloor \frac{x-l}{s} \right\rceil, 0, 2^b - 1\right) + l, \qquad s = \frac{u-l}{2^b - 1}. \tag{2}$$

In conventional layer-wise activation PTQ, a single set of clipping bounds is shared by all tokens at each layer and quantization site:

$$\hat{x}_i^{(\ell,m)} = Q\left(x_i^{(\ell,m)}; l_{\ell,m}, u_{\ell,m}, b\right), \tag{3}$$



(a) Activation distribution.

(b) Layer-wise token-range CoV.

Figure 1: **Selective scan induces token-wise activation heterogeneity.** (a) Post-scan activations exhibit a broader distribution than pre-scan activations in a representative ASSM layer (layer 18). (b) Per-token range CoV increases after selective scan in most ASSM layers.

where ℓ denotes the layer, m denotes the activation quantization site, and i denotes the spatial token. The bounds $(l_{\ell,m}, u_{\ell,m})$ are calibrated offline and fixed during inference. Therefore, layer-wise activation PTQ is effective only when one set of bounds can represent the activation distributions of all tokens at the same quantization site.

4 Observations

This section analyzes how selective scan affects token-wise activation distributions in Mamba-based restoration backbones. We study 20 ASSM blocks of MambaIRv2-Light and collect pre-scan and post-scan activations from FP32 ASSM modules using low-resolution DIV2K [38] validation patches.

4.1 Selective Scan Induces Token-wise Activation Heterogeneity

The central question for low-bit activation quantization is whether a single set of layer-wise clipping bounds can represent all tokens at the same quantization site. In Mamba-based restoration backbones, selective scan undermines this assumption by producing token-dependent activation regimes.

Distributional expansion. Fig. 1(a) shows that post-scan activations spread over a broader range than pre-scan activations after median-$|x|$ normalization. This indicates that selective scan disperses tokens into different magnitude regimes, making one clipping range poorly matched to all tokens.

Per-token range variation. For each token i, we define its activation range as $r_i = \max_j x_{i,j} - \min_j x_{i,j}$ and measure token-wise dispersion by

$$\mathrm{CoV}(r) = \frac{\mathrm{std}_i(r_i)}{\mathrm{mean}_i(r_i)}. \tag{4}$$

As shown in Fig. 1(b), post-scan CoV exceeds pre-scan CoV in **18 out of 20** ASSM layers, with an average amplification of about $1.80\times$. Thus, token-wise range variation is a consistent effect of selective scan rather than a layer-specific artifact.

Conflict with low-bit quantization. With a single layer-wise clipping bound, fitting high-response tokens enlarges the step size and wastes resolution for typical tokens, whereas fitting typical tokens severely clips high-response tokens. This trade-off becomes more severe as the bit-width decreases. The issue is therefore not isolated outliers, but token populations with distinct activation ranges. This motivates considering group-wise clipping bounds for low-bit Mamba SR.

4.2 From Observation to Design

Section 4.1 shows that selective scan makes a single layer-wise clipping range poorly matched to all tokens. However, larger token-wise variation alone does not guarantee useful groups. For low-bit PTQ, groups should be defined by *quantization compatibility*: tokens should share a clipping range only when they prefer similar downstream rounding–clipping trade-offs.

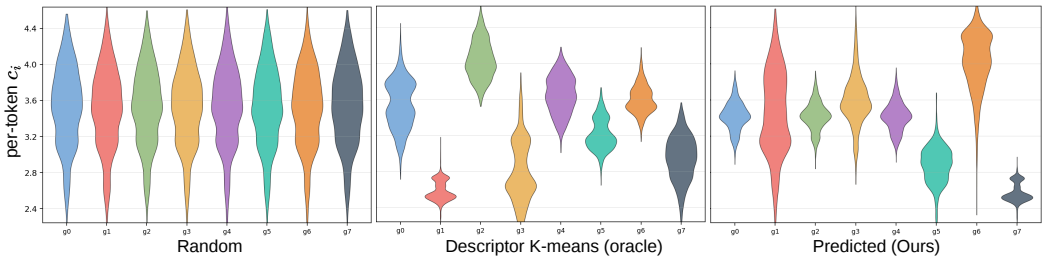


Figure 2: **Clipping-scale regimes.** Violin plots show token-level preferred clipping scales $c_i^\star$ under random, post-scan descriptor K-means (oracle), and predicted groups.

Fig. 2 visualizes this criterion using the token-level preferred clipping scale $c_i^\star = \arg\min_c \|x_i - Q(x_i; -c, c, b)\|_2^2$, a symmetric proxy for the clipping range preferred by token i. Random grouping yields similar clipping-scale distributions across groups, whereas post-scan descriptor K-means separates tokens into distinct clipping-scale regimes. The pre-scan predicted groups recover part of this hardware-unfriendly oracle structure without observing downstream activations.

The most direct realization is a *downstream oracle* that assigns groups from post-scan activations, but this is hardware-unfriendly for static low-bit deployment because it requires runtime statistics inside the quantized path. A static approximation must therefore satisfy two constraints: **C1**, group indices must be produced by a frozen function without runtime statistics; and **C2**, group indices must be predicted before the target grouped quantization sites while being supervised by labels derived from their post-scan activation behavior.

These constraints rule out simpler alternatives. Layer-wise PTQ ignores token heterogeneity; runtime post-scan grouping violates C1; per-token dynamic clipping requires irregular scale metadata; and pre-scan clustering is deployable but not aligned with the downstream rounding–clipping trade-off, violating C2. Thus, we use pre-scan group prediction, downstream-aware targets, and fixed clipping-bound tables. This is feasible because pre-scan features generate token-dependent selective-scan parameters and therefore provide the earliest deployable signal for approximating the oracle.

5 Method: Token-Group Quantization

Section 4.2 motivates a static approximation to the post-scan grouping oracle. Token-Group Quantization implements this design with three components: a group-conditional activation quantizer, downstream-aware pseudo-labels, and a lightweight pre-scan predictor. At inference, the frozen predictor selects entries from fixed clipping-bound tables without runtime statistics or post-scan group assignment. Fig. 3 illustrates the pipeline.

5.1 Group-Conditional Activation Quantizer

We apply group-wise clipping bounds to the activation sites where token-dependent distributions are most pronounced. In each block, this includes eight group-conditional sites: three downstream linear sites and five scan-internal per-token operands. The remaining quantized sites use layer-wise PTQ.

Given the predicted group assignment $\hat{g}_i^{(\ell)}$, token i at layer ℓ and quantization site m is quantized as

$$\hat{x}_i^{(\ell,m)} = Q\left(x_i^{(\ell,m)}; l_{\ell,m,\hat{g}_i^{(\ell)}}, u_{\ell,m,\hat{g}_i^{(\ell)}}, b\right). \tag{5}$$

The bounds $(l_{\ell,m,g}, u_{\ell,m,g})$ form a layer-, site-, and group-specific clipping-bound table.

5.2 Quantization-Aware Pseudo-Labels

A pre-scan predictor requires supervision, but the labels should encode quantization-error regimes rather than semantics. We therefore construct pseudo-labels offline from downstream activation descriptors, so that tokens in the same group prefer similar clipping bounds at the quantization sites.

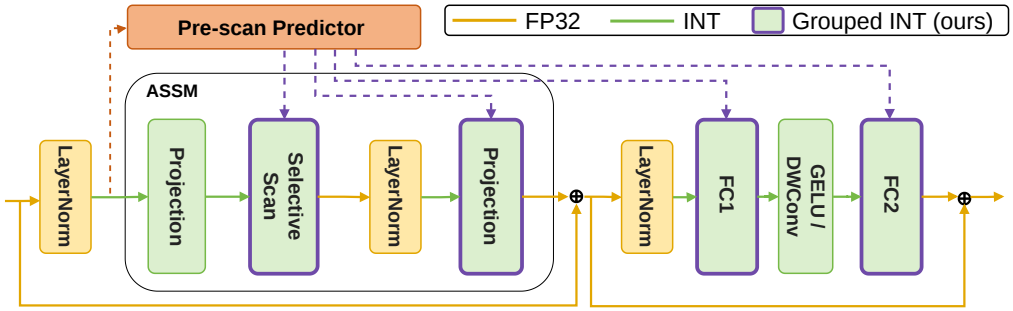


Figure 3: Overview of Token-Group Quantization. A frozen pre-scan predictor predicts token groups, and grouped INT sites use offline-calibrated clipping-bound tables without runtime statistics.

For each token and group-conditional site $m \in \mathcal{M}$, where $\mathcal{M}$ denotes the eight sites above, we use a descriptor summarizing the rounding–clipping trade-off:

$$\phi_m\big(x_i^{(\ell,m)}\big) = \Big[q_{0.99}(|x_i^{(\ell,m)}|),\ q_{0.999}(|x_i^{(\ell,m)}|),\ \mathrm{RMS}(x_i^{(\ell,m)}),\ \max|x_i^{(\ell,m)}|\Big]. \tag{6}$$

Here, q_p is the empirical p-quantile over channel entries. For each token–layer pair (i,ℓ), we concatenate descriptors over $m \in \mathcal{M}$. We then run a single K-means over the pooled token–layer samples from the calibration set to obtain pseudo-labels $g_i^{\star(\ell)}$. Descriptor construction and clustering are used only offline. See Appendix C.

5.3 Pre-scan Group Prediction

Following C1 and C2, group indices must be predicted from pre-scan inputs without runtime activation statistics. Let $\tilde{x}_i^{(\ell)}$ be the pre-scan feature of token i at layer ℓ. Since selective scan generates token-dependent recurrence parameters from these features, we implement the predictor as a shared two-layer FFN:

$$\hat{g}_i^{(\ell)} = \arg\max_{k\in\{1,\dots,K\}} \Big[f_\theta(\tilde{x}_i^{(\ell)})\Big]_k. \tag{7}$$

Although f_θ is shared, $\tilde{x}_i^{(\ell)}$ is layer-dependent, so the assignment remains layer-adaptive. We refer to this as structure–magnitude decoupling: the shared FFN captures recurring token-structure cues for grouping, whereas the layer- and site-specific bound tables absorb the actual magnitude drift induced by selective scan. See Appendix D. The predictor is trained from pre-scan inputs by

$$\mathcal{L}_{\mathrm{pred}} = \mathbb{E}_{(i,\ell)}\left[\mathrm{CE}\left(f_\theta(\tilde{x}_i^{(\ell)}), g_i^{\star(\ell)}\right)\right]. \tag{8}$$

5.4 Bound Initialization and Teacher Distillation

After training and freezing the pre-scan predictor, we replay the calibration set and collect activations according to the predicted group assignment $\hat{g}_i^{(\ell)}$. For each (ℓ, m, g), we form the activation pool assigned to group g:

$$\mathcal{X}_{\ell,m}(g) = \{x_i^{(\ell,m)} : \hat{g}_i^{(\ell)} = g\}. \tag{9}$$

Then we initialize the clipping bounds by minimizing empirical quantization error:

$$(l_{\ell,m,g}, u_{\ell,m,g}) = \arg\min_{l,u} \mathbb{E}_{x\sim\mathcal{X}_{\ell,m}(g)}\left[\|x - Q(x;l,u,b)\|_2^2\right]. \tag{10}$$

Starting from these bounds, we refine only the clipping-bound tensors using a full-precision teacher. Given teacher/student outputs $\mathbf{y}, \hat{\mathbf{y}}$ and features $F_h, \hat{F}_h$ at distillation layers $h \in \mathcal{H}$, we use

$$\mathcal{L}_{\mathrm{pix}} = \|\hat{\mathbf{y}} - \mathbf{y}\|_1, \qquad \mathcal{L}_{\mathrm{feat}} = \frac{1}{|\mathcal{H}|}\sum_{h\in\mathcal{H}}\left\|\frac{\hat{F}_h}{\|\hat{F}_h\|_2} - \frac{F_h}{\|F_h\|_2}\right\|_2^2. \tag{11}$$

Table 1: Quantitative comparison under the Weight-freeze PTQ protocol on MambaIRv2-Light.

Scale	Method	W / A	Set5		Set14		BSD100		Urban100		Manga109	
			PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
×4	MambaIRv2-Light	32 / 32	32.51	0.8992	28.84	0.7878	27.75	0.7426	26.82	0.8079	31.24	0.9182
	PTQ4VM [8] (WACV'25)	4 / 4	30.82	0.8670	27.69	0.7546	26.95	0.7115	24.76	0.7321	28.19	0.8660
	Quamba [6] (ICLR'25)	4 / 4	31.01	0.8715	27.77	0.7585	26.99	0.7149	25.01	0.7470	28.57	0.8752
	MambaQuant [46] (ICLR'25)	4 / 4	30.74	0.8650	27.17	0.7413	26.37	0.6920	23.28	0.6694	26.73	0.8186
	OAQ-SR [40] (ICCV'25)	4 / 4	30.40	0.8577	27.41	0.7470	26.75	0.7045	24.31	0.7136	27.39	0.8485
	Ours	**4 / 4**	**31.46**	**0.8811**	**28.14**	**0.7683**	**27.24**	**0.7231**	**25.39**	**0.7600**	**29.21**	**0.8884**
	PTQ4VM [8] (WACV'25)	2 / 2	28.77	0.8162	26.36	0.7167	26.16	0.6802	23.37	0.6704	25.26	0.7943
	Quamba [6] (ICLR'25)	2 / 2	28.88	0.8080	26.45	0.7131	26.20	0.6752	23.48	0.6651	25.43	0.7818
	MambaQuant [46] (ICLR'25)	2 / 2	28.84	0.8079	26.41	0.7114	26.18	0.6739	23.45	0.6648	25.38	0.7829
	OAQ-SR [40] (ICCV'25)	2 / 2	28.78	0.8122	26.37	0.7143	26.17	0.6780	23.40	0.6681	25.33	0.7892
	Ours	**2 / 2**	**29.12**	**0.8260**	**26.57**	**0.7259**	**26.32**	**0.6891**	**23.57**	**0.6805**	**25.58**	**0.8031**
×2	MambaIRv2-Light	32 / 32	38.26	0.9615	34.09	0.9221	32.36	0.9019	33.26	0.9378	39.35	0.9785
	PTQ4VM [8] (WACV'25)	4 / 4	37.17	0.9549	32.86	0.9099	31.57	0.8900	30.47	0.9084	37.22	0.9706
	Quamba [6] (ICLR'25)	4 / 4	37.07	0.9544	32.77	0.9092	31.47	0.8896	30.54	0.9107	36.94	0.9699
	MambaQuant [46] (ICLR'25)	4 / 4	36.67	0.9495	31.76	0.8899	30.85	0.8756	28.08	0.8407	33.47	0.9186
	OAQ-SR [40] (ICCV'25)	4 / 4	37.22	0.9551	32.60	0.9071	31.51	0.8897	30.49	0.9122	36.91	0.9717
	Ours	**4 / 4**	**37.56**	**0.9578**	**33.24**	**0.9149**	**31.86**	**0.8950**	**31.34**	**0.9204**	**37.85**	**0.9742**
	PTQ4VM [8] (WACV'25)	2 / 2	34.38	0.9328	31.05	0.8886	30.21	0.8660	27.61	0.8603	32.04	0.9399
	Quamba [6] (ICLR'25)	2 / 2	34.66	0.9339	31.26	0.8899	30.38	0.8687	27.80	0.8613	32.50	0.9407
	MambaQuant [46] (ICLR'25)	2 / 2	34.65	0.9337	31.22	0.8885	30.36	0.8685	27.78	0.8610	32.43	0.9395
	OAQ-SR [40] (ICCV'25)	2 / 2	34.54	0.9332	31.19	0.8907	30.33	0.8705	27.71	0.8613	32.29	0.9408
	Ours	**2 / 2**	**35.02**	**0.9436**	**31.41**	**0.8966**	**30.51**	**0.8766**	**27.90**	**0.8708**	**32.67**	**0.9508**

5.5 Frequency-Preserving Refinement for Bound Calibration

Low-bit activation quantization can attenuate textures and shift edges, while pixel-space distillation is often dominated by low-frequency energy. Motivated by the effectiveness of frequency-domain objectives in image restoration [14, 7], we formulate Frequency-Preserving Refinement (FPR) as a teacher-supervised objective for SR PTQ. FPR calibrates clipping bounds by aligning Fourier amplitude and phase between full-precision teacher and quantized student outputs.

We apply a 2D Hann window w and compute 2D Fourier transforms $\mathbf{Y} = \mathcal{F}(\mathbf{y} \odot w)$ and $\hat{\mathbf{Y}} = \mathcal{F}(\hat{\mathbf{y}} \odot w)$, then decompose coefficients into amplitude $|\mathbf{Y}|$ and phase $\angle\mathbf{Y} \in [-\pi, \pi]$:

$$\mathcal{L}_{\mathrm{FPR}} = \tfrac{1}{2} \underbrace{\big\| \, |\mathbf{Y}| - |\hat{\mathbf{Y}}| \, \big\|_1}_{\text{amplitude}} + \tfrac{1}{2} \underbrace{\big\| \, \mathrm{wrap}(\angle\mathbf{Y} - \angle\hat{\mathbf{Y}}) \, \big\|_1}_{\text{phase}}, \tag{12}$$

where $\mathrm{wrap}(\cdot)$ maps the phase difference to $(-\pi, \pi]$. Amplitude matching preserves high-frequency energy, while phase matching penalizes structural shifts in edges and textures.

Combining the three distillation terms gives the final objective:

$$\mathcal{L} = \mathcal{L}_{\mathrm{pix}} + \lambda_{\mathrm{feat}} \, \mathcal{L}_{\mathrm{feat}} + \lambda_{\mathrm{FPR}} \, \mathcal{L}_{\mathrm{FPR}}. \tag{13}$$

Only clipping-bound tensors are optimized, while backbone weights, pseudo-labels, predictor parameters, and bit-widths remain fixed.

6 Experiments

6.1 Experimental Setup

Datasets. We follow the standard SR PTQ protocol in prior work [39, 30]. Calibration is performed using only low-resolution images without high-resolution targets. We construct the calibration set from 96 low-resolution patches sampled from the DF2K training split [38, 29]. Evaluation is conducted on Set5 [3], Set14 [47], BSD100 [34], Urban100 [19], and Manga109 [35].

Metrics and protocol. We report PSNR and SSIM [41] on the Y channel in YCbCr. Following common SR practice, image boundaries are cropped by the upscaling factor before evaluation.

Backbone. We use MambaIRv2-Light [17] as the primary backbone. We additionally evaluate MambaIRv1 [16] to verify the token-group quantization generalizes across selective-scan SR architectures.

Table 3: Quantitative comparison under the Weight-training PTQ with LoRA protocol on MambaIRv2-Light.

Scale	Method	W / A	Set5		Set14		BSD100		Urban100		Manga109	
			PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
×4	MambaIRv2-Light	32 / 32	32.51	0.8992	28.84	0.7878	27.75	0.7426	26.82	0.8079	31.24	0.9182
	AIQViT [21] (AAAI'25)	4 / 4	31.51	0.8856	28.10	0.7707	27.26	0.7268	25.18	0.7546	28.76	0.8854
	SPR^{2}Q [45] (ICLR'26)	4 / 4	32.04	0.8929	28.59	0.7812	27.52	0.7359	26.07	0.7864	30.12	0.9048
	Ours + LoRA	**4 / 4**	**32.25**	**0.8953**	**28.67**	**0.7830**	**27.61**	**0.7377**	**26.24**	**0.7906**	**30.49**	**0.9095**
	AIQViT [21] (AAAI'25)	2 / 2	30.52	0.8667	27.44	0.7541	26.79	0.7102	24.35	0.7210	27.27	0.8515
	SPR^{2}Q [45] (ICLR'26)	2 / 2	30.43	0.8635	27.45	0.7531	26.86	0.7110	24.37	0.7215	27.30	0.8529
	Ours + LoRA	**2 / 2**	**30.93**	**0.8749**	**27.80**	**0.7626**	**27.06**	**0.7190**	**24.72**	**0.7365**	**28.11**	**0.8709**
×2	MambaIRv2-Light	32 / 32	38.26	0.9615	34.09	0.9221	32.36	0.9019	33.26	0.9378	39.35	0.9785
	AIQViT [21] (AAAI'25)	4 / 4	37.73	0.9598	33.19	0.9148	31.92	0.8968	30.97	0.9167	37.59	0.9750
	SPR^{2}Q [45] (ICLR'26)	4 / 4	37.84	0.9601	33.42	0.9171	31.99	0.8972	32.09	0.9286	38.60	0.9772
	Ours + LoRA	**4 / 4**	**38.02**	**0.9608**	**33.77**	**0.9200**	**32.24**	**0.9005**	**32.41**	**0.9314**	**38.76**	**0.9777**
	AIQViT [21] (AAAI'25)	2 / 2	36.99	0.9552	32.35	0.9092	31.42	0.8907	29.70	0.9007	36.01	0.9700
	SPR^{2}Q [45] (ICLR'26)	2 / 2	36.92	0.9527	32.38	0.9105	31.46	0.8910	30.12	0.9058	36.98	0.9707
	Ours + LoRA	**2 / 2**	**37.27**	**0.9580**	**32.93**	**0.9121**	**31.72**	**0.8939**	**30.34**	**0.9090**	**37.02**	**0.9729**

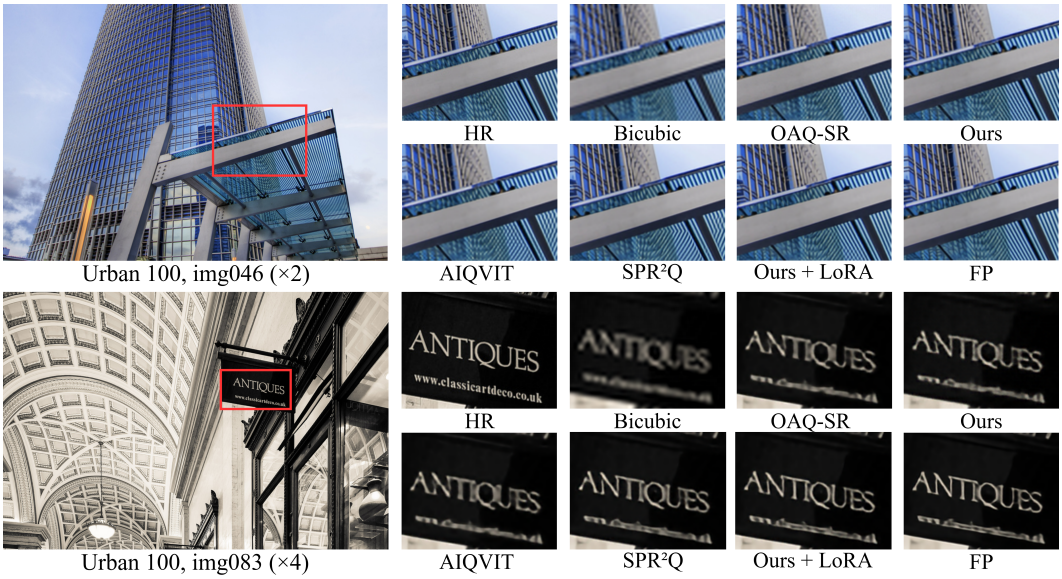


Figure 4: Visual comparison for image SR (×2 and ×4) on Urban100 under 4-bit PTQ.

Implementation details. Calibration images are cropped and augmented following prior SR PTQ work [45, 30]. We initialize clipping bounds by empirical quantization-error minimization (Eq. 10) and refine the clipping bounds using Eq. 13 with λ_{feat}=5 and λ_{FPR}=0.01. Unless otherwise specified, we use K=8 token groups. We use a shared two-layer FFN predictor detailed in Appendix E.1. Bound refinement is performed for 3,000 iterations with a batch size of 8. All experiments are conducted on three NVIDIA RTX A6000 GPUs. Reported numbers use seed 0, and repeating the full pipeline under five seeds gives std below 0.03 dB at W4A4 and 0.05 dB at W2A2 (Appendix F.1).

Baselines. For the Weight-freeze PTQ protocol, we compare with PTQ4VM [8], Quamba [6], MambaQuant [46], and OAQ-SR [40]. PTQ4VM, Quamba, and MambaQuant results are taken from SPR^{2}Q [45], while OAQ-SR and ours are evaluated by us. For the Weight-training PTQ with LoRA protocol, we compare with AIQViT [21] and SPR^{2}Q [45]. Our Ours+LoRA follows the SPR^{2}Q rectifier training-and-fusion protocol, while applying token-group activation quantization. All methods use the same W/A low-bit scope. Table 2 shows that this low-bit path covers 90.8% of FLOPs, where Quant. for scan operands denotes activation-operand quantization.

Table 2: FLOPs and W/A quantization scope.

Module	FLOPs (G)	Ratio (%)	Quant.
Linear + Conv	8.49	85.9	Yes
Scan operands	0.49	4.9	Yes
Other Conv	0.90	9.1	No
Total	9.88	100.0	–

Table 4: Calibration cost on MambaIRv2-Light ($\times 2$, W4A4).

Method	Wall-clock	Iter	Params
Weight-freeze PTQ			
OAQ-SR	1.29 h	3,000	1.9 k
Ours	1.42 h	3,000	4.1 k
Weight-training PTQ with LoRA			
SPR^{2}Q	6.12 h	12,500	617.2 k
Ours + LoRA	6.57 h	12,500	619.4 k

Table 5: Latency on Set5, MambaIRv2-Light ($\times 2$, W4A4).

Method	Latency (ms)	Speedup
NVIDIA RTX A6000		
FP16	180.7	1.00×
4-bit Oracle	254.5	0.71×
4-bit Base	102.7	1.76×
4-bit Ours	106.2	1.70×
Jetson Orin Nano		
FP16	3149.6	1.00×
4-bit Oracle	3213.9	0.98×
4-bit Base	1380.1	2.28×
4-bit Ours	1410.7	2.23×

Table 6: Number of token groups K (MambaIRv2-Light, $\times 2$, W4A4).

K	Set14 PSNR / SSIM	Urban100 PSNR / SSIM
1	32.62 / 0.9073	30.59 / 0.9127
4	33.07 / 0.9117	31.03 / 0.9175
6	33.19 / 0.9142	31.28 / 0.9199
8	**33.24 / 0.9149**	**31.34 / 0.9204**

Table 7: Supervision ablation on MambaIRv2-Light at $\times 2$ and W4A4. TG marks token-group quantization. Rows differ only in the refinement objective.

Refinement objective	TG	Set14 PSNR / SSIM	Urban100 PSNR / SSIM
None (Eq. 10 init. only)	✗	32.15 / 0.8931	29.84 / 0.8915
$\mathcal{L}_{\text{pix}}+\mathcal{L}_{\text{feat}}$ (layer-wise PTQ)	✗	32.45 / 0.9056	30.34 / 0.9107
None (Eq. 10 init. only)	✓	32.38 / 0.9015	30.56 / 0.9115
$\mathcal{L}_{\text{pix}}$ only	✓	32.53 / 0.9018	30.72 / 0.9123
$\mathcal{L}_{\text{pix}}+$ feat. MSE (conventional KD)	✓	33.01 / 0.9091	30.89 / 0.9136
$\mathcal{L}_{\text{pix}}+\mathcal{L}_{\text{feat}}$	✓	33.06 / 0.9107	30.99 / 0.9172
+ amplitude term only	✓	33.15 / 0.9122	31.14 / 0.9190
+ phase term only	✓	33.10 / 0.9128	31.10 / 0.9183
+ $\mathcal{L}_{\text{FPR}}$ (amp. + phase, **Ours**)	✓	**33.24 / 0.9149**	**31.34 / 0.9204**

Table 8: Grouping rule ablation (MambaIRv2-Light, $\times 2$, W4A4). Static: no runtime statistics (C1). The post-scan oracle (gray) is a hardware-unfriendly upper bound; bold marks the best static rule.

Rule	Static	Set14 PSNR / SSIM	Urban100 PSNR / SSIM
Layer-wise PTQ	✓	32.45 / 0.9056	30.34 / 0.9107
Random groups	✓	31.76 / 0.8904	29.91 / 0.9070
Pre-scan K-means	✓	32.84 / 0.9092	30.77 / 0.9161
Post-scan oracle	✗	33.41 / 0.9176	31.57 / 0.9253
Predicted (Ours)	✓	**33.24 / 0.9149**	**31.34 / 0.9204**

6.2 Results

Quantitative results. Table 1 shows that our Weight-freeze PTQ consistently outperforms Mamba-specific and SR-oriented PTQ baselines across both scales and bit-widths. Table 3 evaluates the LoRA-style Weight-training protocol. Our Ours+LoRA variant follows the rectifier training-and-fusion protocol of SPR^{2}Q while using our token-group activation quantization. It achieves the best results in all settings and nearly recovers full precision at 4-bit; on BSD100, the PSNR drop is only 0.12 dB for $\times 2$ and 0.14 dB for $\times 4$.

Qualitative results. Figure 4 shows that our method better preserves structural details under low-bit quantization. In the $\times 2$ example, Ours maintains the straight canopy edges closer to the FP result, while in the $\times 4$ example, Ours reconstructs sharper text contours with less blur than the baselines.

Offline calibration cost. Table 4 compares the calibration cost on MambaIRv2-Light ($\times 2$, 4-bit). In the Weight-freeze PTQ setting, our method adds only a small overhead over OAQ-SR, increasing the wall-clock time from 1.29 h to 1.42 h. Under the LoRA-style Weight-training protocol, Ours+LoRA also remains close to SPR^{2}Q, with 6.57 h vs. 6.12 h. These results show that the proposed group-wise activation quantization introduces only modest additional calibration cost.

Inference speed. Table 5 reports end-to-end latency of the W4A4 static low-bit inference path with byte-aligned integer kernels and fixed quantization codes. Our method achieves 1.70× speedup on A6000 and 2.23× on Jetson Orin Nano over FP16, nearly matching the base 4-bit path, whereas the runtime post-scan oracle loses the speedup entirely since it needs an extra observation pass before quantization. Appendix G explains the overhead of both paths.

6.3 Ablation Study

Number of token groups. Table 6 shows that performance improves as the number of token groups increases, while the gain from $K = 6$ to $K = 8$ nearly saturates. We therefore use $K = 8$ as the default setting. On MambaIRv2-Base the optimum shifts mildly to K=16, but K=8 already recovers 89% of the range (Appendix F.3).

Component analysis. Table 7 separates the two sources of gain. Token grouping alone accounts for 77% of the improvement over layer-wise PTQ on Set14 and 65% on Urban100, and it helps even without refinement. On top of the strongest conventional objective, the amplitude and phase terms

each help alone and their combination adds a further 0.35 dB on Urban100, showing that the two are complementary. Appendix H shows that this gain comes from the high-frequency band that pixel and feature matching under-supervise.

Grouping rule. Table 8 shows that the gain is not from having more ranges: random groups fall below layer-wise PTQ, and our predicted groups outperform pre-scan K-means and recover 82.3% and 81.3% of the oracle gain on Set14 and Urban100 without runtime statistics.

FPR objective weight. A sensitivity analysis on λ_{FPR} is provided in Appendix I.

Generalization. Appendices A and B extend the evaluation to MambaIRv1-Light, MambaIRv1-Base, MambaIRv2-Base, denoising, and JPEG artifact removal, to the Transformer backbone SwinIR-light against 2DQuant, FIMA-Q, APHQ-ViT, and SPR^{2}Q, and to ImageNet classification on Vision Mamba, where the token-range amplification of Sec. 4 is even larger. Ours is best in every setting. Ported Hadamard, SpinQuant, and Affine transforms trail ours by up to 2.12 dB, and SpinQuant combined with ours is best everywhere, so the token axis and the channel axis compose. Appendix F.2 further shows robustness to input resolution up to 16 times the calibration length.

7 Conclusion

We presented Token-Group Quantization, a static yet token-adaptive PTQ framework for low-bit Mamba-based super-resolution. By analyzing MambaIRv2-Light, we showed that selective scan amplifies token-wise activation heterogeneity, making conventional layer-wise clipping ranges poorly matched to all tokens. Our method addresses this mismatch with a lightweight pre-scan predictor, quantization-aware pseudo-labels, and offline-calibrated group-wise clipping bounds. We further introduced a teacher-supervised FPR objective for frequency-preserving clipping-bound refinement. Experiments demonstrate consistent gains over prior Mamba and SR PTQ baselines under Weight-freeze and LoRA-based protocols, transfer to Transformer SR and Vision Mamba, and compose with channel-axis transforms, while preserving practical acceleration on Jetson Orin Nano.

Limitation and Future Work. While Token-Group Quantization improves low-bit Mamba SR accuracy and preserves practical acceleration, its pre-scan prediction and group-wise bound lookup introduce a small inference overhead relative to the base quantized path. Future work may explore deployment on other on-device accelerators, such as NPUs, DSPs, mobile GPUs, or vendor-specific low-bit runtimes, under the same token-adaptive quantization framework.

Acknowledgments and Disclosure of Funding

This work was supported by POSCO, and the Cooperative Research Program for Agriculture Science and Technology Development (Project No. RS-2024-00332198) of the Rural Development Administration, and supported by the Institute of Information Communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2025-02219317, AI Star Fellowship(Kookmin University)), and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (RS-2025-23524783)

References

[1] Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

[2] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. Aciq: Analytical clipping for integer quantization of neural networks. 2018.

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

[4] Xiangyu Chen, Xintao Wang, Wenlong Zhang, Xiangtao Kong, Yu Qiao, Jiantao Zhou, and Chao Dong. Hat: Hybrid attention transformer for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[5] Yujie Chen, Haotong Qin, Zhang Zhang, Michelo Magno, Luca Benini, and Yawei Li. Q-mambair: Accurate quantized mamba for efficient image restoration. *arXiv preprint arXiv:2503.21970*, 2025.

[6] 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.

[7] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021.

[8] Younghyun Cho, Changhun Lee, Seonggon Kim, and Eunhyeok Park. Ptq4vm: Post-training quantization for visual mamba. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1176–1185. IEEE, 2025.

[9] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

[10] Jinwoo Chung, Sangho An, Sungyeop Jung, and Jangho Kim. Taq: Static-deployable temporal-aware quantization for real-world video super-resolution. In *European Conference on Computer Vision*, pages 167–184. Springer, 2026.

[11] Jinwoo Chung, Sungyeop Jung, Weronika Czorapinska, and Jangho Kim. Quantized model soup shake-up: Weight perturbation for enhanced ensemble diversity. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 687–698, 2026.

[12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[13] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

[14] Dario Fuoli, Luc Van Gool, and Radu Timofte. Fourier space losses for efficient perceptual image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2360–2369, 2021.

[15] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024.

[16] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. In *European conference on computer vision*, pages 222–241. Springer, 2024.

[17] Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mambairv2: Attentive state space restoration. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28124–28133, 2025.

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

[19] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015.

[20] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.

[21] Runqing Jiang, Ye Zhang, Longguang Wang, Pengpeng Yu, and Yulan Guo. Aiqvit: Architecture-informed post-training quantization for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17635–17643, 2025.

[22] Jangho Kim, Jayeon Yoo, Yeji Song, KiYoon Yoo, and Nojun Kwak. Finding efficient pruned network via refined gradients for pruned weights. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 9003–9011, 2023.

[23] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[24] Eunji Kwon, Jongho Yoon, and Seokhyeong Kang. Mobile transformer accelerator exploiting various line sparsity and tile-based dynamic quantization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(6):1808–1821, 2023.

[25] Junghyup Lee, Dohyung Kim, and Bumsub Ham. Network quantization with element-wise gradient scaling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6448–6457, 2021.

[26] 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. *arXiv preprint arXiv:2411.05007*, 2024.

[27] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021.

[28] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021.

[29] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[30] Kai Liu, Haotong Qin, Yong Guo, Xin Yuan, Linghe Kong, Guihai Chen, and Yulun Zhang. 2dquant: Low-bit post-training quantization for image super-resolution. *Advances in Neural Information Processing Systems*, 37:71068–71084, 2024.

[31] 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, 2024.

[32] Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In *International Conference on Learning Representations*, volume 2025, pages 92009–92032, 2025.

[33] Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. Affinequant: Affine transformation quantization for large language models. In *International Conference on Learning Representations*, volume 2024, pages 50932–50951, 2024.

[34] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[35] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia tools and applications*, 76(20):21811–21838, 2017.

[36] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International conference on machine learning*, pages 7197–7206. PMLR, 2020.

[37] Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. *arXiv preprint arXiv:2410.09426*, 2024.

[38] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017.

[39] Zhijun Tu, Jie Hu, Hanting Chen, and Yunhe Wang. Toward accurate post-training quantization for image super resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5856–5865, 2023.

[40] Hailing Wang, Jianglin Lu, Yitian Zhang, and Yun Fu. Outlier-aware post-training quantization for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16175–16184, 2025.

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612, 2004.

[42] Zhuguanyu Wu, Shihe Wang, Jiayi Zhang, Jiaxin Chen, and Yunhong Wang. Fima-q: Post-training quantization for vision transformers by fisher information matrix approximation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14891–14900. IEEE, 2025.

[43] Zhuguanyu Wu, Jiayi Zhang, Jiaxin Chen, Jinyang Guo, Di Huang, and Yunhong Wang. Aphq-vit: Post-training quantization with average perturbation hessian based reconstruction for vision transformers. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9686–9695. IEEE, 2025.

[44] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pages 38087–38099. PMLR, 2023.

[45] Jingwei Xin, Wenhao Li, Nannan Wang, Jie Li, and Xinbo Gao. SPR^{2}Q: Static priority-based rectifier routing quantization for image super-resolution. In *The Fourteenth International Conference on Learning Representations*.

[46] Zukang Xu, Yuxuan Yue, Xing Hu, Zhihang Yuan, Zixu Jiang, Zhixuan Chen, Jiangyong Yu, Chen Xu, Sifan Zhou, and Dawei Yang. Mambaquant: Quantizing the mamba family with variance aligned rotation methods. *arXiv preprint arXiv:2501.13484*, 2025.

[47] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010.

[48] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.

[49] Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving. *Proceedings of Machine Learning and Systems*, 6:196–209, 2024.

[50] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

[51] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

A Generalization Protocol

We evaluate generalization on MambaIRv1-Light, JPEG artifact removal, and color image denoising. For these appendix experiments, we include two complementary baselines. *Layer-wise PTQ* uses the same Weight-freeze protocol and quantization scope as ours, but uses one clipping range per layer/site instead of group-wise ranges; thus, it directly isolates the benefit of token grouping. OAQ-SR is included as a recent SR-oriented outlier-aware PTQ baseline. Since the goal of these experiments is to test transferability rather than to establish a new benchmark suite for every restoration task, this combination provides a controlled baseline and a strong task-relevant PTQ baseline.

These experiments also test the robustness of pre-scan group prediction under degraded restoration inputs. JPEG artifact removal contains spatially non-uniform compression artifacts, while color denoising uses noisy inputs; thus, the predictor must infer useful token groups from corrupted features rather than clean or single-degradation inputs. The consistent gains on these tasks indicate that the proposed grouping rule remains effective under input degradation.

Table 9: Generalization results on MambaIRv1-Light under the Weight-freeze PTQ protocol.

Scale	Method	W / A	Set5		Set14		BSD100		Urban100		Manga109	
			PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
×4	MambaIRv1-Light	32 / 32	32.42	0.8977	28.74	0.7847	27.68	0.7400	26.52	0.7983	30.94	0.9135
	Layer-wise PTQ	4 / 4	30.39	0.8257	27.45	0.7249	26.71	0.6811	24.99	0.7185	28.62	0.8372
	OAQ-SR [40] (ICCV'25)	4 / 4	30.85	0.8507	27.80	0.7446	26.93	0.7019	25.03	0.7392	28.59	0.8611
	Ours	**4 / 4**	**31.10**	**0.8627**	**27.91**	**0.7534**	**27.09**	**0.7097**	**25.23**	**0.7444**	**28.90**	**0.8682**
	Layer-wise PTQ	2 / 2	23.10	0.4199	22.29	0.3725	22.25	0.3453	20.92	0.3736	21.99	0.4017
	OAQ-SR [40] (ICCV'25)	2 / 2	27.98	0.7367	25.68	0.6477	25.55	0.6133	23.07	0.6080	25.02	0.7135
	Ours	**2 / 2**	**28.33**	**0.7611**	**26.09**	**0.6713**	**25.89**	**0.6365**	**23.29**	**0.6278**	**25.19**	**0.7390**
×2	MambaIRv1-Light	32 / 32	38.13	0.9610	33.95	0.9208	32.31	0.9013	32.85	0.9349	39.20	0.9782
	Layer-wise PTQ	4 / 4	35.07	0.9178	31.83	0.8774	30.67	0.8581	30.05	0.8835	35.06	0.9325
	OAQ-SR [40] (ICCV'25)	4 / 4	36.55	0.9445	32.67	0.9013	31.46	0.8821	30.58	0.9042	36.53	0.9593
	Ours	**4 / 4**	**37.12**	**0.9527**	**33.06**	**0.9105**	**31.73**	**0.8910**	**31.06**	**0.9160**	**37.25**	**0.9686**
	Layer-wise PTQ	2 / 2	22.57	0.4159	21.95	0.4086	21.80	0.3942	21.33	0.4402	22.47	0.4216
	OAQ-SR [40] (ICCV'25)	2 / 2	34.03	0.9303	30.76	0.8809	29.22	0.8471	27.01	0.8537	31.78	0.9354
	Ours	**2 / 2**	**34.40**	**0.9371**	**31.05**	**0.8896**	**30.20**	**0.8685**	**27.60**	**0.8624**	**32.07**	**0.9436**

Table 10: Generalization results on color image denoising with noise level $\sigma = 15$ under the Weight-freeze PTQ protocol.

Method	W / A	CBSD68		Kodak24		McMaster		Urban100	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
MambaIRv2-Base	32 / 32	34.15	0.9388	35.15	0.9363	35.07	0.9421	34.72	0.9484
Layer-wise PTQ	4 / 4	32.84	0.9161	33.49	0.9098	33.06	0.9169	32.62	0.9229
OAQ-SR [40] (ICCV'25)	4 / 4	33.52	0.9290	34.33	0.9243	34.04	0.9305	33.59	0.9371
Ours	**4 / 4**	**33.78**	**0.9335**	**34.69**	**0.9299**	**34.51**	**0.9356**	**34.04**	**0.9419**
Layer-wise PTQ	2 / 2	27.20	0.7871	27.00	0.7667	24.31	0.7564	25.44	0.7860
OAQ-SR [40] (ICCV'25)	2 / 2	31.63	0.8960	32.06	0.8838	31.16	0.8857	30.25	0.8944
Ours	**2 / 2**	**32.41**	**0.9142**	**32.89**	**0.9060**	**31.94**	**0.9081**	**31.38**	**0.9174**

Table 11: Generalization results on JPEG artifact removal under the Weight-freeze PTQ protocol.

Quality	Method	W / A	Classic5		LIVE1	
			PSNR↑	SSIM↑	PSNR↑	SSIM↑
$q = 10$	MambaIRv2-Base	32 / 32	30.37	0.8269	29.91	0.8301
	Layer-wise PTQ	4 / 4	28.95	0.7834	28.74	0.7928
	OAQ-SR [40] (ICCV'25)	4 / 4	29.59	0.8064	29.26	0.8125
	Ours	**4 / 4**	**29.93**	**0.8183**	**29.57**	**0.8230**
	Layer-wise PTQ	2 / 2	27.57	0.7324	27.48	0.7474
	OAQ-SR [40] (ICCV'25)	2 / 2	28.06	0.7511	28.06	0.7686
	Ours	**2 / 2**	**28.30**	**0.7698**	**28.22**	**0.7851**
$q = 30$	MambaIRv2-Base	32 / 32	33.81	0.8970	33.73	0.9179
	Layer-wise PTQ	4 / 4	32.47	0.8752	32.37	0.8940
	OAQ-SR [40] (ICCV'25)	4 / 4	33.05	0.8871	33.05	0.9084
	Ours	**4 / 4**	**33.39**	**0.8927**	**33.34**	**0.9133**
	Layer-wise PTQ	2 / 2	22.78	0.4427	22.70	0.4525
	OAQ-SR [40] (ICCV'25)	2 / 2	31.60	0.8617	31.45	0.8822
	Ours	**2 / 2**	**31.90**	**0.8719**	**31.90**	**0.8934**
$q = 40$	MambaIRv2-Base	32 / 32	34.64	0.9093	34.73	0.9323
	Layer-wise PTQ	4 / 4	33.34	0.8909	33.28	0.9110
	OAQ-SR [40] (ICCV'25)	4 / 4	34.02	0.9033	34.14	0.9262
	Ours	**4 / 4**	**34.31**	**0.9068**	**34.38**	**0.9292**
	Layer-wise PTQ	2 / 2	30.79	0.8153	31.17	0.8413
	OAQ-SR [40] (ICCV'25)	2 / 2	32.83	0.8895	32.86	0.9110
	Ours	**2 / 2**	**33.11**	**0.8925**	**33.22**	**0.9147**

Tables 9–11 show that our method consistently outperforms both baselines across backbones, tasks, and bit-widths. This suggests that the proposed static token-group activation ranges address a general activation-heterogeneity issue in Mamba restoration models, rather than overfitting to the main MambaIRv2-Light SR setting.

Notably, the same pre-scan prediction principle remains effective even when the input features are degraded by noise or JPEG artifacts, suggesting that the predictor does not rely on clean-image-specific cues but captures quantization regimes that persist across restoration tasks. For generalization experiments, we prioritize controlled isolation over exhaustive benchmarking.

B Generalization Beyond Mamba-Based SR

The main paper evaluates MambaIRv2-Light as the primary backbone, and Appendix A adds MambaIRv1-Light and MambaIRv2-Base on further restoration tasks. This section tests whether the observation of Sec. 4 and the proposed grouping principle extend to a Transformer SR backbone, to Mamba outside restoration, to larger MambaIR backbones, and to channel-axis quantization transforms from the LLM and diffusion literature.

B.1 Transformer SR Backbone

We apply the same framework to SwinIR-light [28] with the identical quantization scope as 2DQuant [30] and SPR^{2}Q [45], replacing the post-attention MLP activation quantizers with our group-conditional quantizers at K=8. Since SwinIR has no scan stage, the predictor becomes a pre-site predictor trained from pre-block features. Group indices are still produced before the target quantization sites without runtime statistics, so the deployment constraint C1 is unchanged. All distillation settings follow the main paper. Table 12 compares against 2DQuant and the ViT PTQ methods FIMA-Q [42] and APHQ-ViT [43] under the Weight-freeze protocol, and Table 13 compares against SPR^{2}Q under the LoRA protocol. Ours is best in every cell.

On SwinIR-light we measured token-range CoV before and after attention and did not observe the amplification induced by selective scan. Token ranges still vary with image content, so tokens can be clustered into distinct range regimes and both the descriptor K-means and the pre-site predictor operate as designed. Since the inter-group range gap is smaller without scan amplification, the clipping error recoverable by grouping is also smaller, which is consistent with the more modest margins than on MambaIRv2-Light. The gain magnitude thus tracks the degree of token heterogeneity across backbones. The method is generally applicable and particularly beneficial for Mamba.

Table 12: $\times 2$ SR on SwinIR-light under the Weight-freeze PTQ protocol.

Method	W / A	Set5		Set14		BSD100		Urban100		Manga109	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
SwinIR-light	32 / 32	38.14	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.12	0.9783
2DQuant (NeurIPS'24)	4 / 4	37.87	0.9594	33.41	0.9161	32.02	0.8971	31.84	0.9251	38.31	0.9761
FIMA-Q (CVPR'25)	4 / 4	37.71	0.9598	33.38	0.9160	32.01	0.8964	31.95	0.9283	38.44	0.9764
APHQ-ViT (CVPR'25)	4 / 4	37.76	0.9591	33.41	0.9165	31.94	0.8965	31.98	0.9280	38.34	0.9759
Ours	**4 / 4**	**37.91**	**0.9600**	**33.47**	**0.9171**	**32.09**	**0.8978**	**32.06**	**0.9287**	**38.51**	**0.9773**
2DQuant (NeurIPS'24)	2 / 2	36.00	0.9497	31.98	0.9012	30.91	0.8810	28.62	0.8819	34.40	0.9602
FIMA-Q (CVPR'25)	2 / 2	36.06	0.9515	32.10	0.9048	31.01	0.8848	28.77	0.8873	34.75	0.9638
APHQ-ViT (CVPR'25)	2 / 2	36.14	0.9517	32.14	0.9049	31.04	0.8850	28.86	0.8885	34.99	0.9644
Ours	**2 / 2**	**36.58**	**0.9530**	**32.36**	**0.9056**	**31.17**	**0.8853**	**29.18**	**0.8908**	**35.47**	**0.9657**

Table 13: $\times 2$ SR on SwinIR-light under the Weight-training PTQ with LoRA protocol.

Method	W / A	Set5		Set14		BSD100		Urban100		Manga109	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
SwinIR-light	32 / 32	38.14	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.12	0.9783
SPR^{2}Q (ICLR'26)	4 / 4	37.95	0.9602	33.55	0.9178	32.14	0.8989	32.04	0.9275	38.59	0.9769
Ours + LoRA	**4 / 4**	**38.04**	**0.9608**	**33.65**	**0.9193**	**32.22**	**0.9003**	**32.28**	**0.9302**	**38.75**	**0.9776**
SPR^{2}Q (ICLR'26)	2 / 2	37.28	0.9572	32.83	0.9113	31.63	0.8921	30.20	0.9077	37.08	0.9726
Ours + LoRA	**2 / 2**	**37.69**	**0.9596**	**33.22**	**0.9157**	**31.96**	**0.8971**	**31.16**	**0.9193**	**37.74**	**0.9753**

B.2 Vision Mamba on ImageNet

To test the observation outside restoration, we evaluate Vision Mamba [51] on ImageNet classification. The observation of Sec. 4 transfers. Token-range CoV increases after the Mamba block in all 24 blocks of both models, with an average amplification of $4.94\times$ on Vim-Tiny and $3.11\times$ on Vim-Small, which is larger than the $1.80\times$ measured on MambaIRv2-Light.

For quantization we follow the scope and the 256-image calibration protocol of PTQ4VM [8] and apply our group-conditional quantizers with $K{=}8$. The predictor runs once per block from the pre-scan block input, and the same group indices are reused across all six per-token linear sites in the block. Since classification outputs logits, we replace the pixel loss with a logit KD loss and drop the FPR term while keeping feature distillation. Only the activation clipping bounds are trained. Table 14 shows that ours is best in every setting.

Table 14: ImageNet Top-1 accuracy in percent on Vision Mamba under PTQ. Calibration and quantization scope follow PTQ4VM.

Method	W / A	Vim-Tiny	Vim-Small
FP	32 / 32	76.07	80.48
Layer-wise PTQ	6 / 6	71.92	76.66
PTQ4VM (WACV'25)	6 / 6	74.42	79.62
Ours	**6 / 6**	**75.88**	**79.95**
Layer-wise PTQ	4 / 4	53.61	67.06
PTQ4VM (WACV'25)	4 / 4	56.62	70.41
Ours	**4 / 4**	**57.98**	**72.50**

Although the token-range dispersion is even larger on Vim than in restoration, its impact is task-dependent. Classification aggregates token features into a single prediction, whereas restoration is dense per-pixel regression under static deployment, where every token is on the critical path with all ranges fixed ahead of time. The dispersion therefore becomes the first-order bottleneck in restoration. At aggressive bit-widths it resurfaces in classification as well, where ours again improves over PTQ4VM.

B.3 Larger MambaIR Backbones

Table 15 evaluates MambaIRv1-Base and MambaIRv2-Base at $\times 2$ under the Weight-freeze protocol with the same quantization scope, calibration setup, and baselines as Appendix A. Ours is best in every setting, so the failure mode identified in Sec. 4 and its remedy are not confined to lightweight models.

Table 15: $\times 2$ SR on MambaIRv1-Base and MambaIRv2-Base under the Weight-freeze PTQ protocol.

Backbone	Method	W / A	Set14		Urban100	
			PSNR↑	SSIM↑	PSNR↑	SSIM↑
MambaIRv2-Base	FP	32 / 32	34.89	0.9275	34.49	0.9468
	Layer-wise PTQ	4 / 4	32.39	0.9060	29.25	0.8930
	OAQ-SR (ICCV'25)	4 / 4	33.04	0.9132	31.25	0.9195
	Ours	**4 / 4**	**33.28**	**0.9157**	**31.60**	**0.9236**
	Layer-wise PTQ	2 / 2	30.54	0.8556	27.53	0.8398
	OAQ-SR (ICCV'25)	2 / 2	31.33	0.8958	27.81	0.8699
	Ours	**2 / 2**	**31.47**	**0.8974**	**28.13**	**0.8756**
MambaIRv1-Base	FP	32 / 32	34.67	0.9261	34.15	0.9446
	Layer-wise PTQ	4 / 4	31.72	0.8808	29.07	0.8719
	OAQ-SR (ICCV'25)	4 / 4	32.19	0.8952	29.53	0.8874
	Ours	**4 / 4**	**33.07**	**0.9094**	**31.17**	**0.9153**
	Layer-wise PTQ	2 / 2	29.13	0.7846	25.47	0.6770
	OAQ-SR (ICCV'25)	2 / 2	30.84	0.8830	27.43	0.8561
	Ours	**2 / 2**	**31.23**	**0.8940**	**27.71**	**0.8672**

B.4 Relation to Channel-Axis Quantization Transforms

Rotation, affine, and scaling quantizers from the LLM and diffusion literature [1, 32, 33, 37, 44, 26] share one property with prior SR and Mamba PTQ methods. The quantization range is assigned independently of token identity, through a channel-axis transform shared by every token. This is sound when token distributions are homogeneous, which is exactly the assumption that selective scan breaks in Sec. 4. To verify this empirically, we ported the core transforms onto MambaIRv2-Light under the identical protocol, sites, calibration set, and budget. Hadamard covers QuaRot [1], SpinQuant [32] covers learned rotation, and Affine covers AffineQuant [33] and FlatQuant [37], with SmoothQuant-style scale migration [44] as its diagonal case. Atom [49] alone acts on the token axis but through runtime statistics, which C1 rules out, so ours is its static counterpart.

Table 16 shows two findings. First, even the best channel-axis transform on each benchmark trails ours by up to 2.12 dB at W4A4. Second, the two axes compose. SpinQuant combined with our method is best in every setting, with the largest gains at W2A2. The two remove different error sources, so Token-Group Quantization occupies a design axis these literatures do not cover while remaining compatible with them.

Table 16: $\times 2$ SR on MambaIRv2-Light with channel-axis transforms ported onto the identical Weight-freeze PTQ protocol. Affine covers AffineQuant and FlatQuant, and Hadamard covers QuaRot.

Method	W / A	Set5		Set14		BSD100		Urban100		Manga109	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
MambaIRv2-Light	32 / 32	38.26	0.9615	34.09	0.9221	32.36	0.9019	33.26	0.9378	39.35	0.9785
Affine	4 / 4	35.80	0.9452	32.81	0.9091	30.89	0.8789	30.57	0.9163	34.67	0.9550
Hadamard	4 / 4	36.33	0.9492	32.61	0.9085	31.04	0.8823	30.36	0.9123	35.73	0.9655
SpinQuant	4 / 4	36.42	0.9500	32.50	0.9063	31.08	0.8828	30.74	0.9188	35.54	0.9622
Ours	4 / 4	37.56	0.9578	33.24	0.9149	31.86	0.8950	31.34	0.9204	37.85	0.9742
SpinQuant + Ours	**4 / 4**	**37.69**	**0.9588**	**33.40**	**0.9163**	**31.96**	**0.8965**	**31.61**	**0.9228**	**37.94**	**0.9750**
Affine	2 / 2	34.59	0.9365	31.18	0.8904	30.32	0.8703	27.71	0.8643	32.31	0.9438
Hadamard	2 / 2	34.72	0.9389	31.23	0.8923	30.37	0.8722	27.75	0.8656	32.35	0.9456
SpinQuant	2 / 2	34.84	0.9400	31.29	0.8935	30.42	0.8734	27.79	0.8668	32.46	0.9467
Ours	2 / 2	35.02	0.9436	31.41	0.8966	30.51	0.8766	27.90	0.8708	32.67	0.9508
SpinQuant + Ours	**2 / 2**	**36.23**	**0.9513**	**32.17**	**0.9037**	**31.06**	**0.8841**	**28.77**	**0.8850**	**34.72**	**0.9614**

C Rationale for Quantization-Aware Descriptors

C.1 Why These Descriptor Statistics?

We motivate the descriptor in Eq. 6 from the error decomposition of a uniform quantizer. For clarity, consider symmetric clipping $[-c, c]$ for a token-level random variable X_i drawn from the channel entries of $x_i^{(\ell,m)}$. The quantization step size is $\Delta(c) = 2c/(2^b - 1)$. Under the standard high-resolution approximation, the per-token quantization error can be written as

$$E_i(c) \approx \frac{\Delta(c)^2}{12} \Pr(|X_i| \le c) + \mathbb{E}\left[(|X_i| - c)_+^2\right]. \tag{14}$$

Here, $(z)_+ = \max(z, 0)$. The first term is the rounding error within the clipping range and increases with the step size, while the second term is the clipping error from tail activations. Therefore, tokens should share a clipping bound only when their bulk scale and tail behavior are similar.

The descriptor in Eq. 6 provides a compact proxy for this precision–clipping trade-off. $\mathrm{RMS}(x_i^{(\ell,m)})$ captures the bulk activation energy that influences rounding sensitivity, $q_{0.99}$ and $q_{0.999}$ capture near-tail and extreme-tail scales, and $\max|x_i^{(\ell,m)}|$ captures the worst-case tail extent. Although these statistics are not sufficient statistics for the full activation distribution, they characterize the dominant factors that shape the quantization-error curve over clipping bounds. Clustering tokens in this descriptor space therefore groups tokens that are likely to have similar MSE-preferred clipping ranges.

D Pre-scan Predictor Details and Robustness

D.1 Prediction Accuracy and Error Structure

The pre-scan predictor is trained to reproduce pseudo-labels obtained by K-means clustering in the downstream activation-descriptor space. For $K = 8$, the predictor achieves approximately **84%** top-1 accuracy on held-out tokens. This accuracy should not be interpreted as classification accuracy against semantic ground truth. The pseudo-labels are calibration artifacts that compress quantization-error regimes, so the accuracy measures how well pre-scan features recover the descriptor-space partition used for grouping.

More importantly, prediction errors are structured. We further observe that many misclassified tokens are assigned to nearby clusters in the descriptor space rather than to distant regimes. Since the descriptor is constructed from statistics that dominate the rounding–clipping trade-off, nearby clusters tend to have similar MSE-preferred clipping bounds. Therefore, such errors still often select bounds that are close to the oracle-preferred range. This supports the robustness of using a lightweight pre-scan predictor instead of runtime post-scan group assignment.

The pseudo-labels should therefore be understood not as semantic labels, but as a compact representation of quantization-error regimes. What matters is not perfect label stability, but whether tokens assigned to the same group exhibit similar clipping-error and rounding-error behavior.

D.2 Robustness to Inference-Time Predictor Errors

While Table 8 compares different grouping rules—each requiring its own calibration—we now ask a complementary question: how much does the framework tolerate *runtime* errors in the predictor output? In Table 17, we keep the calibrated bound tables and the trained predictor exactly as in our default deployment, and only at evaluation time randomly reassign a fraction ρ of the predicted group ids to a uniformly random group in $\{1, \ldots, K\}$. No re-calibration, no bound re-refinement, and no predictor re-training is performed. The ρ=0% row therefore matches our default result, while ρ=100% corresponds to fully random group ids *on top of* the bound tables calibrated for our predictor, and is conceptually distinct from the "Random groups" row of Table 8, where bounds are calibrated jointly with random grouping.

Table 17: **Robustness to inference-time predictor perturbation** on MambaIRv2-Light ($\times$2, W4A4) under the Weight-freeze PTQ protocol. We use the calibrated bound tables and the trained predictor from our default setting and, only at evaluation time, randomly reassign a fraction ρ of predicted group ids to a uniformly random group in $\{1, \ldots, K\}$. Calibration, bound refinement, and the predictor are unchanged across rows. The ρ=0% row matches our default deployment.

Perturbation rate ρ	Set14		Urban100	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
$\rho = 0\%$ (ours, no perturbation)	33.24	0.9149	31.34	0.9204
$\rho = 5\%$	33.21	0.9145	31.34	0.9204
$\rho = 10\%$	33.18	0.9143	31.33	0.9202
$\rho = 20\%$	33.09	0.9137	31.25	0.9193
$\rho = 30\%$	33.01	0.9127	31.13	0.9178
$\rho = 50\%$	32.72	0.9108	31.02	0.9165
$\rho = 100\%$ (uniform random ids)	32.56	0.9084	30.45	0.9121

The empirical predictor error rate is approximately 16% (K=8, top-1 accuracy 84%). Around this regime, perturbation has only a marginal effect: at ρ=10% and ρ=20%, the Set14 PSNR drops by only 0.06 dB and 0.15 dB from the unperturbed default, with similarly small drops on Urban100. Even at ρ=50%—where half of all predicted group ids are randomly corrupted—PSNR degrades by only 0.52 dB on Set14 and 0.32 dB on Urban100, well below the gap between layer-wise PTQ and our method (Table 8).

A particularly informative comparison is the ρ=100% row, in which predicted group ids are entirely replaced by uniformly random ids while the bound tables remain those calibrated for our predictor. This setting still attains 32.56 / 30.45 dB on Set14 / Urban100, clearly above the "Random groups" row of Table 8 (31.76 / 29.91 dB), where the bound tables themselves were calibrated under random grouping. The 0.80 / 0.54 dB gap indicates that a substantial portion of the benefit comes from the calibrated bound tables themselves, with the predictor providing additional gains on top.

Together with the structured-error analysis in Appendix D, this confirms that the framework does not rely on the predictor being near-perfect. Small fluctuations in predicted group ids are absorbed by the bound tables, and the method degrades gracefully rather than catastrophically as prediction quality decreases.

E Additional Implementation Details

E.1 Pre-scan Predictor Training Details

The pre-scan predictor is a shared two-layer FFN ($C \to 64 \to K$ with GELU, where C=48 for MambaIRv2-Light and K=8 by default). It is trained only during calibration to reproduce the descriptor-K-means pseudo-labels using cross-entropy. We use Adam with learning rate 10^{-3}, batch size 1024, 20 epochs, an 80/20 train/validation split, and random seed 0. The input is the pre-ASSM feature at each token–layer pair, and the target is the K-means label obtained from descriptors concatenated over all eight group-conditional sites: three downstream linear sites and five scan-internal

per-token operands. After training, the FFN is frozen and used only to assign group indices for fixed clipping-bound table lookup.

E.2 Quantization Scope

We apply low-bit quantization to two functional groups of activations. First, we quantize the inputs to the linear and pointwise convolution modules in the backbone, including the post-scan projection `out_proj` and the two `convffn2` linear layers in each ASSM block. This part accounts for 85.9% of the FLOPs. Second, inside selective scan, we quantize five token-adaptive operands: `xs`, `dts`, `delta`, `Bs`, and `Cs`. These scan operands account for an additional 4.9% of the FLOPs. Together, the low-bit path covers 90.8% of the total FLOPs, matching the quantization scope reported in Table 2.

Among these quantized activations, group-wise clipping is applied to the eight token-indexed sites used for pseudo-label construction: `out_proj`, `convffn2.fc1`, `convffn2.fc2`, and the five scan operands `xs`, `dts`, `delta`, `Bs`, and `Cs`. The remaining modules are kept in full precision due to sensitivity or limited benefit from low-bit quantization.

For feature distillation, $\mathcal{H}$ contains the outputs of all ASSM blocks. In MambaIRv2-Light, $|\mathcal{H}| = 20$, and $F_h, \hat{F}_h$ denote the corresponding full-precision teacher and quantized student features after the residual output of block h.

E.3 Reproducibility

The source code is publicly available at `https://github.com/imaboybut/TGQ-MambaSR`.

F Robustness Analyses

F.1 Sensitivity to Calibration Sampling and Random Seeds

The calibration pipeline involves patch sampling, K-means initialization, predictor training, and bound refinement. We repeat the full pipeline from scratch under five random seeds on MambaIRv2-Light at $\times 2$. Each seed jointly re-samples the 96 patches from DF2K, re-initializes the K-means, and re-seeds predictor training. Bound initialization and refinement are re-run per seed, so nothing is shared except the pretrained backbone.

The results are highly stable. The standard deviation across seeds stays below 0.03 dB PSNR and 0.005 SSIM on all five benchmarks at W4A4, and below 0.05 dB and 0.01 at W2A2. These deviations are far smaller than the margins over the strongest baseline, such as the 0.85 dB gain over OAQ-SR on Urban100 at W4A4, so the improvements are not a favorable calibration draw. The predictor reaches $85 \pm 1.3\%$ top-1 accuracy across runs, close to the 84% reported in Appendix D, which indicates that the pseudo-labels remain similarly learnable across seeds. This stability is consistent with the design, since the bound tables absorb group-assignment fluctuations and such small swings are well within the tolerance shown in Table 17.

F.2 Robustness to Input Resolution

Calibration uses 64×64 LR crops of about 4k tokens, while selective scan runs unwindowed over $L{=}HW$, so Urban100 evaluation already reaches about 308k tokens, which is 75 times the calibration length. To isolate resolution from content, Table 18 freezes the predictor and bound tables and crops the same Urban100 images across a 64 times token range. We report the gap between FP and Ours, since crop size shifts content statistics for both models identically and the gap isolates what the frozen grouping rule loses. The gap remains nearly flat across the whole range, and at 16 times the calibration length it is no larger than at the calibration length itself. The method is therefore insensitive to input resolution, and this flatness under fixed $K{=}8$ suggests that per-resolution retuning of K is unnecessary.

Table 18: Resolution robustness on Urban100 at $\times 2$ and W4A4 over 20 images. Predictor and bound tables are frozen and only the LR input crop size varies. Gap is FP minus Ours.

LR crop	Tokens	Ratio to calib.	FP		Ours		Gap	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
32×32	1,024	$0.25\times$	33.77	0.9127	31.88	0.8929	1.89	0.0198
64×64	4,096	$1\times$	32.60	0.9084	30.48	0.8825	2.12	0.0259
128×128	16,384	$4\times$	31.89	0.9105	29.78	0.8873	2.11	0.0232
256×256	65,536	$16\times$	32.13	0.9217	30.07	0.9003	2.06	0.0214

F.3 Number of Token Groups on a Larger Backbone

Table 19 repeats the K sweep of Table 6 on MambaIRv2-Base, extended to $K{=}12$ and $K{=}16$. The optimum shifts mildly upward with model size, since a wider channel dimension supports more distinguishable activation regimes, but the returns diminish quickly. Moving from $K{=}1$ to $K{=}8$ already recovers 89% of the total range on Set14. $K{=}8$ therefore remains a robust default without per-model tuning, and larger backbones can raise K for a small further gain. The identical two-layer FFN transfers without architectural change, and only its input width follows the backbone channel dimension.

Table 19: Effect of the number of token groups K on MambaIRv2-Base at $\times 2$ and W4A4.

K	Set14		Urban100	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
1	32.55	0.9099	30.98	0.9084
4	33.20	0.9141	31.37	0.9193
8, default	33.28	0.9157	31.60	0.9236
12	33.34	0.9163	31.68	0.9244
16	**33.37**	**0.9168**	**31.71**	**0.9249**

G Deployment Details

We implement a CUDA deployment path for the W4A4 quantized model to evaluate practical 4-bit acceleration. After calibration, all clipping-bound tables, quantization scales, weight quantization parameters, and pre-scan predictor parameters are frozen. Runtime execution predicts token groups from pre-scan features and selects the corresponding fixed clipping bounds, without runtime statistics collection, dynamic clipping updates, or post-scan group assignment.

For grouped activation sites, calibration-time fake-quantization modules are replaced with static CUDA kernels that fuse group-wise bound lookup, clamping, rounding, and dequantization. The activation and weight quantization codes are constrained to the 4-bit range, while byte-aligned integer buffers are used for efficient CUDA execution. Quantized linear layers use pre-quantized weights, integer CUDA kernels, and dequantization with fixed activation and weight quantization parameters.

We also apply CUDA graph capture after static conversion to reduce CPU dispatch overhead. As reported in Table 5, the W4A4 deployment achieves $1.70\times$ speedup on A6000 and $2.23\times$ speedup on Jetson Orin Nano over FP16, demonstrating practical 4-bit acceleration for Token-Group Quantization.

We measure end-to-end latency on Set5 with batch size 1 using the same evaluation script, backend, graph mode, and timing protocol for FP16, the 4-bit base path, and our grouped 4-bit path. For each image, we run 5 warm-up iterations, synchronize the device, run 10 timed repetitions, synchronize again, and report the average latency per image. Since Set5 contains images with different resolutions, CUDA graph capture is performed once for each encountered input shape and then reused for that shape. The reported latency excludes graph construction time but includes pre-scan predictor execution, group-wise bound lookup, quantization/dequantization, and low-bit linear execution.

Overhead breakdown. The measured latency in Table 5 includes every component of the grouped path. Predictor execution is inside the timed path, and group-wise bound lookup, clamping, rounding,

and dequantization are fused into the static CUDA kernels described above. The resulting overhead over the base 4-bit path is 3.4% on A6000 and 2.2% on Jetson Orin Nano, and the added memory is about 5 KB of bound tables plus a two-layer FFN, so the two paths are nearly identical in cost. By hardware-friendly we mean the property that matters for static deployment. There are no runtime statistics and no dynamic clipping updates, and all parameters are fixed after calibration, so the path stays compatible with ahead-of-time compilation and CUDA graph capture.

Runtime post-scan oracle. The 4-bit Oracle rows of Table 5 measure the most favorable dynamic implementation of the post-scan oracle of Table 8. Descriptor statistics of Eq. 6 are computed by dedicated kernels and groups are assigned by nearest-centroid lookup against the offline K-means centroids, with no K-means re-run and with the same CUDA graph capture and timing protocol. Its cost has three sources. First, and dominant, the block must run twice. The group id of a token is determined by statistics of its post-scan activations at the eight grouped sites, but those activations exist only after the block has executed, and the quantizers of the same block already need the id. The only resolution is to run the block once to observe, assign groups, and run it again quantized. Our predictor breaks this dependency by reading pre-scan features, so one pass suffices. Second, the descriptors themselves are costly. At 160 sites, eight per block over 20 blocks, the activation tensor is read once more to obtain per-token $q_{0.99}$, $q_{0.999}$, RMS, and max. The two quantiles are order statistics that cannot be fused into the quantization kernel, so this step adds separate memory-bound kernels and a synchronization point between the two passes. Third, nearest-centroid assignment over the 32-dimensional descriptor is cheap but adds one more launch per block. As a result, the oracle takes 254.5 ms on A6000 and 3213.9 ms on Jetson Orin Nano, a 0.71× and 0.98× speedup over FP16, so 4-bit execution brings no acceleration at all, whereas the base path and ours reach 1.76× and 1.70× on A6000 and 2.28× and 2.23× on Orin Nano. The oracle's 0.17 dB advantage in Table 8 therefore comes at a cost that static low-bit deployment cannot pay, whereas ours retains most of that gain with a single pass and fixed bound tables.

H Frequency-Domain Evidence for FPR

Table 7 in the main paper shows that the amplitude and phase terms each help alone and are complementary in combination. Here we show where in the spectrum the gain arises. Table 20 decomposes the error of the quantized output against the FP output on Urban100 at ×2 and W4A4 into low, mid, and high radial frequency bands and reports the mean amplitude and phase error per band. Set14 shows the same pattern. The variants correspond to the last four rows of Table 7, and the row labeled none is $\mathcal{L}_{\text{pix}}+\mathcal{L}_{\text{feat}}$ with token groups.

Table 20: Band-wise Fourier errors of the quantized output against the FP output on Urban100 at ×2 and W4A4. Lower is better. All variants use token groups, and none denotes $\mathcal{L}_{\text{pix}}+\mathcal{L}_{\text{feat}}$ without any frequency term.

Frequency term	Amplitude error			Phase error		
	Low	Mid	High	Low	Mid	High
none	0.0208	0.2853	0.536	0.0213	0.3211	0.673
+ amplitude only	0.0206	0.2784	**0.512**	0.0212	0.3191	0.668
+ phase only	0.0213	0.2830	0.531	0.0216	0.3181	**0.635**
+ $\mathcal{L}_{\text{FPR}}$, Ours	0.0214	0.2850	0.518	0.0219	0.3213	0.643

Three observations follow. First, the error concentrates in the high band, while low-band errors are an order of magnitude smaller and flat across variants, so pixel and feature matching already cover low frequencies and under-supervise high ones. Second, in the high band the amplitude term primarily reduces amplitude error and the phase term primarily reduces phase error, so the two emphasize complementary error axes. Third, each single term concentrates its supervision on one axis, whereas FPR reduces both high-band errors jointly. Since restoration quality depends on the joint error, only FPR reaches the best PSNR and SSIM in Table 7.

Existing frequency-domain losses [14, 7] train network weights and are supervised by ground-truth HR images or a larger teacher. FPR differs on both sides. Only the clipping bounds are trained while

all weights stay frozen, and the teacher is the FP model supervising its own quantized counterpart. The objective is thus not to learn a representation but to correct the spectral error that quantization itself introduces.

I FPR Objective Weight Sensitivity

Table 21 studies the sensitivity to λ_{FPR} in Eq. 13 under the Weight-freeze PTQ protocol at $\times 2$, W4A4. All other settings, including the pre-scan predictor, bound initialization, $\lambda_{\mathrm{feat}}{=}5$, and the bound-refinement schedule, are kept at their default values. A small weight (0.005) provides too weak a frequency-domain signal, while a large weight (0.1) competes with $\mathcal{L}_{\mathrm{pix}}$ and $\mathcal{L}_{\mathrm{feat}}$ and degrades high-frequency textures, as reflected by the larger drop on Urban100. The performance is stable in a broad neighborhood around the default $\lambda_{\mathrm{FPR}}{=}0.01$.

Table 21: **Sensitivity to FPR objective weight** λ_{FPR} on MambaIRv2-Light ($\times 2$, W4A4) under the Weight-freeze PTQ protocol. All other settings are kept at their default values. $\lambda_{\mathrm{FPR}}{=}0.01$ corresponds to our default setting.

λ_{FPR}	Set14		Urban100	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
0.005	33.11	0.9124	31.19	0.9184
0.01 (ours)	33.24	0.9149	31.34	0.9204
0.05	33.22	0.9140	31.31	0.9197
0.1	33.09	0.9116	31.04	0.9175

J Clipping-Scale Regimes Across Layers

To complement Fig. 2, we visualize token-level preferred clipping scales $c_i^\star$ across additional ASSM layers and post-scan sites. These plots show that downstream descriptor groups often reveal stronger clipping-scale structure than random groups, while predicted groups provide a deployable but less sharply separated approximation.

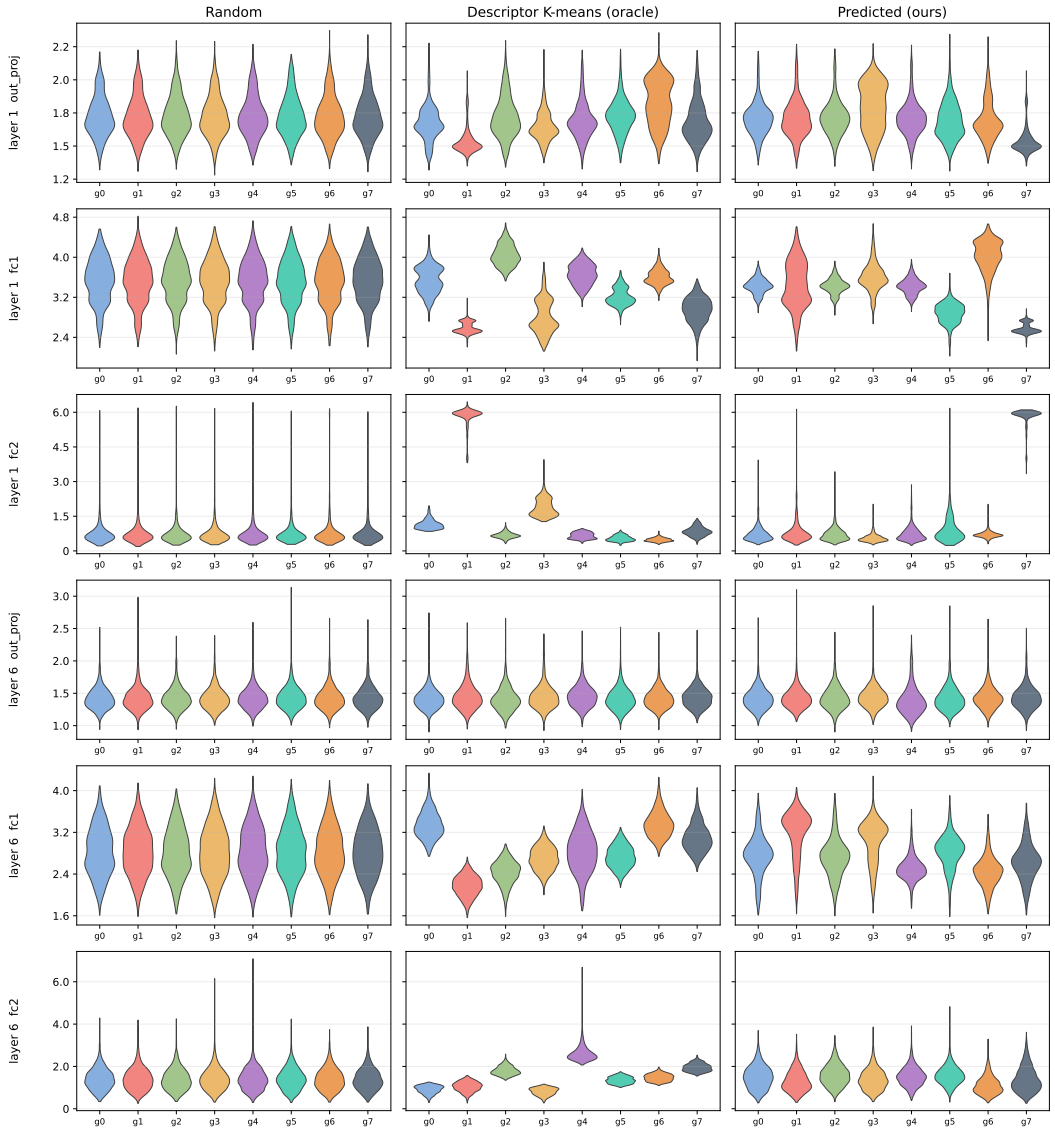


Figure 5: **Clipping-scale regimes in early/middle layers.** Token-level preferred clipping scales $c_i^\star$ for layers 1 and 6 under random, post-scan descriptor K-means (oracle), and predicted groups.

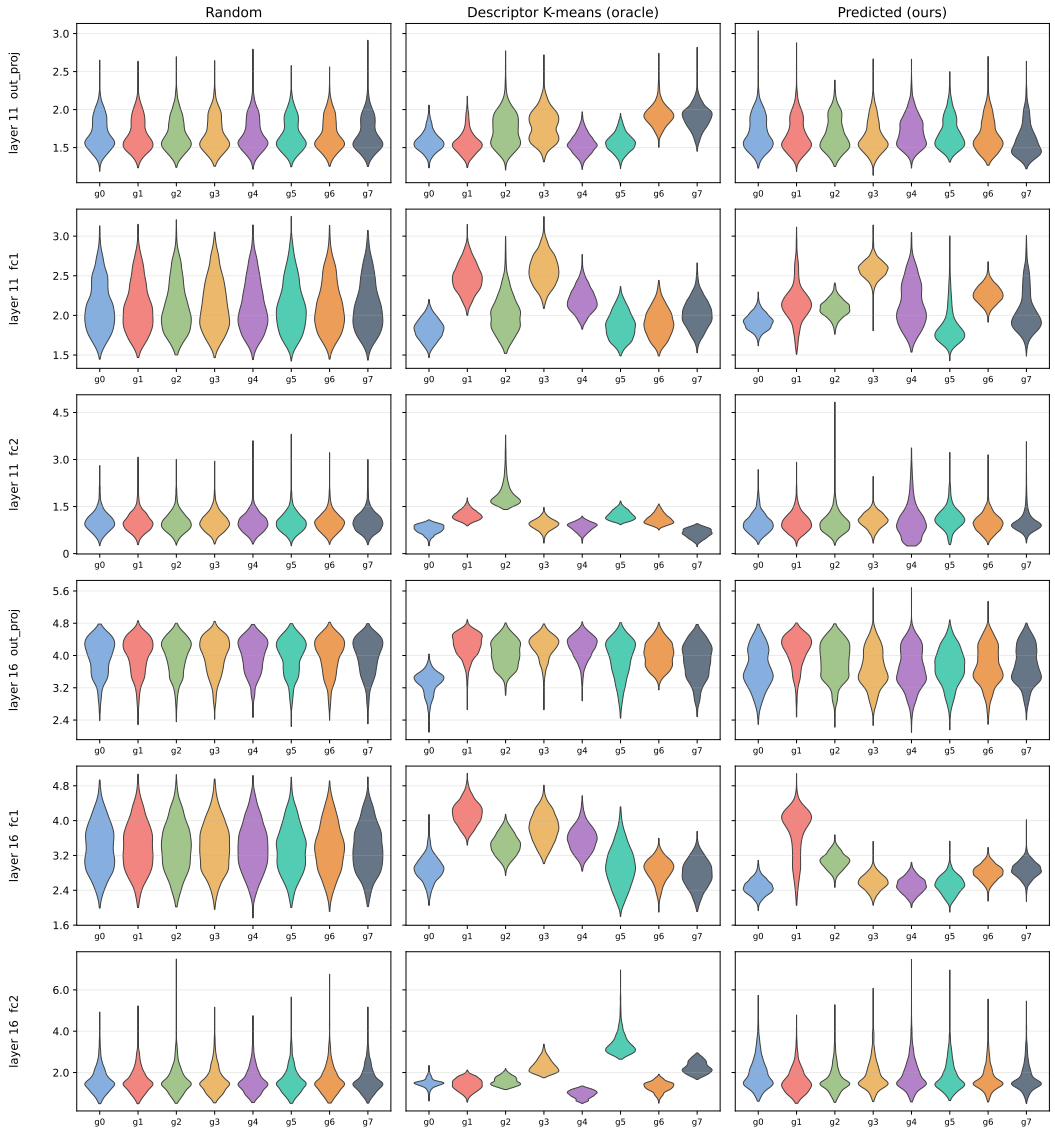


Figure 6: **Clipping-scale regimes in later layers.** Token-level preferred clipping scales $c_i^\star$ for layers 11 and 16 under random, post-scan descriptor K-means (oracle), and predicted groups.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction summarize the main contributions and scope, which are supported by our analysis, ablations, experiments, and latency results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: We discuss limitations and future work in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

Justification: The paper does not present formal theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: We provide dataset, calibration, evaluation, quantization, implementation, and hardware details in the main paper and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes].

Justification: We use public benchmarks and provide the source code and reproduction instructions.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Experimental settings, hyperparameters, calibration details, and predictor training details are provided in the main paper and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

   Justification: We do not report error bars. However, we report the standard deviation over five independent calibration runs on MambaIRv2-Light at $\times 2$ under W4A4 and W2A2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: We report the GPUs, calibration time, iteration counts, trainable parameters, and latency on A6000 and Jetson Orin Nano.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The research uses public image restoration benchmarks and does not involve human subjects, private data, or deceptive data collection.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: The method can reduce edge deployment cost, while efficient image restoration may also be misused in surveillance or misleading image enhancement.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper does not release high-risk models, scraped datasets, or systems with direct high-risk misuse potential.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: We use publicly available SR benchmarks (Set5, Set14, BSD100, Urban100, Manga109, DIV2K, DF2K) and pretrained MambaIRv1/v2 backbones, all under standard research-use licenses, and cite the original papers in the References. Baseline implementations follow the licenses of their public releases.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The source code and its documentation are publicly available.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs are not used as an important or non-standard component of the core research method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.