QuantDemoire: Quantization with Outlier Aware for Image Demoiréing

Zheng Chen[1]*, **Kewei Zhang**[1]*, **Xiaoyang Liu**[1], **Weihang Zhang**[2], **Mengfan Wang**[2], **Yifan Fu**[2], **Yulun Zhang**[1]†
[1]Shanghai Jiao Tong University, [2]Central Media Technology Institute, Huawei

Abstract

Demoiréing aims to remove moiré artifacts that often occur in images. While recent deep learning-based methods have achieved promising results, they typically require substantial computational resources, limiting their deployment on edge devices. Model quantization offers a compelling solution. However, directly applying existing quantization methods to demoiréing models introduces severe performance degradation. The main reasons are distribution outliers and weakened representations in smooth regions. To address these issues, we propose QuantDemoire, a post-training quantization framework tailored to demoiréing. It contains two key components. **First**, we introduce an outlier-aware quantizer to reduce errors from outliers. It uses sampling-based range estimation to reduce activation outliers and keeps a few extreme weights in FP16 with negligible cost. **Second**, we design a frequency-aware calibration strategy. It emphasizes low- and mid-frequency components during fine-tuning, which mitigates banding artifacts caused by low-bit quantization. Experiments validate that our QuantDemoire achieves large reductions in parameters and computation while maintaining quality. Meanwhile, it outperforms existing quantization methods by over **4 dB** on W4A4. We implement a CUDA kernel for the mixed-precision branch and achieve a 2.3× end-to-end speedup on an RTX 4090 GPU. Code is released at: https://github.com/zhengchen1999/QuantDemoire.

1 Introduction

Image demoiréing is a low-level vision task that removes colored stripes caused by frequency aliasing between display screens and imaging sensors. These artifacts reduce image quality and hinder later analysis and recognition. Due to the complex and diverse forms of moiré patterns, effective demoiréing remains a highly challenging task [26, 16, 27].

In recent years, with the rapid progress of deep learning, neural networks have achieved remarkable performance in demoiréing [2, 39, 36, 33, 18]. They adapt to the diverse moiré patterns and achieve excellent performance. However, they usually involve millions of parameters and heavy computation. This limits their use on devices with extremely limited computing power, such as drones, portable cameras and IoT sensors [2, 39]. These devices are also the most important application scenarios of image demoiréing. Therefore, efficient and high-quality demoiréing is critical for practical usage.

Quantization [21, 3, 34] is an effective solution to this challenge. It compresses weights and activations from 32-bit floats to low-bit (*e.g.*, 2∼8 bit) integers. This significantly reduces storage overhead and replaces costly floating-point operations with efficient integer operations. Therefore, quantization speeds up inference and lowers power use, making it well-suited for resource-limited devices. Currently, quantization is widely applied to vision tasks [6, 15, 23].

However, demoiréing lacks dedicated quantization methods. For extremely limited compute and memory, demoiréing models must be compressed to ultra-low bit widths (*e.g.*, 4-bit). Image demoiréing is

*Equal contribution.
†Corresponding author: Yulun Zhang, yulun100@gmail.com

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

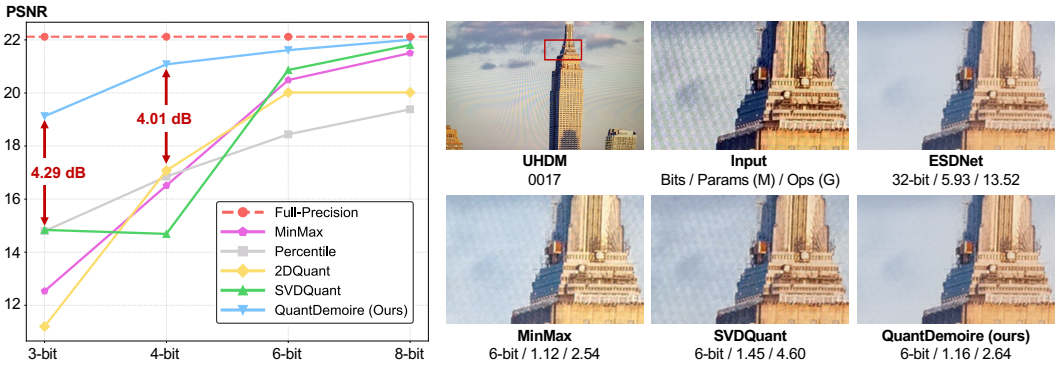

Figure 1: Comparison of PSNR with other quantization methods on UHDM. The full-precision backbone is ESDNet [36]. Our QuantDemoire demonstrates superior performance.

particularly sensitive to such quantization errors because it must suppress false high-frequency moiré patterns while preserving real textures. As a result, directly applying existing quantization [10, 13] results in severe performance loss. The degradation has two causes. **First**, weights and activations outliers expand the quantization range. This reduces the precision of typical values and weakens model's performance. **Second**, quantization weakens representations in smooth images regions. It introduces banding artifacts in mid- and low-frequency components. This not only harms demoiréing performance but also damages details in clean areas. Therefore, it is crucial to design task-specific quantization strategies tailored to the characteristics of the image demoiréing task.

To address these issues, we propose QuantDemoire, a post-training quantization (PTQ) framework for image demoiréing tasks. Our core idea reduces the accuracy loss via the quantizer and the calibration. **(1) Quantizer.** We design the outlier-aware quantizer to reduce the impact of outliers. It uses random sampling to estimate and remove activation outliers. It also keeps a tiny fraction of extreme weights in FP16, with negligible overhead. **(2) Calibration.** We introduce frequency-aware calibration. It emphasizes mid- and low-frequency components when optimizing quantization boundaries. This reduces the banding artifacts in smooth regions caused by quantization.

Extensive experiments show our QuantDemoire significantly improves efficiency while maintaining strong performance. It consistently outperforms existing quantization methods across datasets and bit-widths. As shown in Fig. 1, at 4-bit, our method cuts parameters and computation by over **86.6%** with under **4.7%** PSNR drop vs. FP16. Besides, QuantDemoire surpasses SOTA quantization methods by over **4 dB** (PSNR, 4-bit) on UHDM [36]. Our main contributions are as follows:

- We propose QuantDemoire, the first quantization framework tailored to image demoiréing. It reduces model overhead while preserving restoration ability.
- We design the outlier-aware quantizer that mitigates quantization errors by handling outliers in activations and weights with random sampling and FP16 branch.
- We propose the frequency-aware calibration strategy to optimize mid- and low-frequency features, alleviating quantization-induced banding artifacts.
- We validate our method on multiple datasets and bit-widths. QuantDemoire consistently outperforms existing quantization methods quantitatively and visually.

2 Related Work

2.1 Image demoiréing.

Photos of digital screens often contain colored moiré stripes. These patterns can greatly degrade image quality. Early moiré pattern removal methods used traditional mathematical methods, such as matrix decomposition [16] and spectral models [26]. These techniques generally perform poorly when dealing with diverse types of moiré patterns. In recent years, with the rapid development of deep learning, methods [2, 8, 39, 9, 40, 36, 33, 32, 18], based on deep neural networks, have been proposed. MopNet [8] utilizes multi-scale feature aggregation and attribute-aware classifiers to handle complex frequency. FHDe^{2}Net [9] employs a cascaded architecture to remove multi-scale moiré patterns. ESDNet [36] adds a semantic-aligned, scale-aware module to further improve robustness to scale variations. However, these methods have millions of parameters. The high cost makes them unsuitable for edge devices with limited resources.

2.2 Model Quantization.

Quantization is widely used in neural network compression. It effectively reduces memory and computation by converting 32-bit weights and activations to low-bit integers. Quantization algorithms are generally divided into quantization-aware training (QAT) [6, 1, 23] and post-training quantization (PTQ) [10, 19, 24, 13, 41, 35]. QAT can be accurate [21], but its high training cost limits practical use. PTQ is lightweight and efficient because it requires no retraining or labeled data. SmoothQuant [31] shifts the quantization difficulty from activations to weights. SVDQuant [13] decomposes weights into a low-rank component. Nevertheless, these quantization methods suffer from severe performance degradation on demoiréing models. The influence of outliers [4] and information loss from quantization degrade quality, causing color banding and poor moiré removal. Appendix H.6 further discusses the differences between QuantDemoire and smoothing or mixed-precision quantization methods.

3 Method

In this section, we present our quantization framework, QuantDemoire, for efficient demoiréing (see Fig. 2). We introduce two key components. The first component is the outlier-aware quantizer, which reduces the quantization error introduced by outliers in activations and weights. The second is the frequency-aware calibration strategy, which selectively emphasizes low- and mid-frequency components to mitigate the banding artifacts during the calibration phase.

3.1 Preliminaries: Quantization

We first introduce fake quantization [10], which is employed to simulate quantization errors. Given a weight or activation tensor value v and a quantization bit-width b, the procedure is expressed as:

$$\begin{aligned} \Delta = \frac{u-l}{2^b-1}, \quad z = \text{clip}(\text{Round}(-\frac{l}{\Delta}), 0, 2^b-1), \\ v_z = \text{clip}(\text{Round}(\frac{v}{\Delta}) + z, 0, 2^b-1), \end{aligned} \tag{1}$$

where l and u denote the lower and upper quantization bounds, Δ is the scaling factor, z is the zero-point offset, and v_z is the integer-coded value of v. The clipping function is defined as $\text{clip}(v, l, u) = \max(\min(v, u), l)$, while $\text{Round}(\cdot)$ maps an input to its nearest integer. The corresponding dequantization process can be formulated as:

$$Q(v) = \Delta \times (v_z - z), \tag{2}$$

where $Q(\cdot)$ denotes the operation of fake quantization. Given the input activation $\mathbf{X}$, weight $\mathbf{W}$, and bias $\mathbf{b}$, the output feature $\mathbf{Y}$ can be obtained through:

$$\mathbf{Y} = Q(\mathbf{X}) \cdot Q(\mathbf{W}) + \mathbf{b}. \tag{3}$$

Because $\text{Round}(\cdot)$ is non-differentiable, following prior works [23, 13], the straight-through estimator (STE) is adopted to approximate its gradient. The process is defined as:

$$\frac{\partial Q(v)}{\partial v} \approx \begin{cases} 1 & \text{if } v \in [l, u], \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Moreover, SmoothQuant [31] is widely adopted in quantization to address the challenge arising from the variation in maximum magnitudes across channels. It applies a per-channel scaling transformation to redistribute the quantization difficulty from activations to weights. Given an activation $\mathbf{X}$ and a weight tensor $\mathbf{W}$ for the j-th input channel, the smoothing factor s_j is defined as:

$$\begin{aligned} s_j = \max(|\mathbf{X}_j|)^\alpha / \max(|\mathbf{W}_j|)^{1-\alpha}, \\ \mathbf{s} = (s_1, s_2, \ldots, s_n), \end{aligned} \tag{5}$$

where α is a balancing hyperparameter controlling the trade-off between activation and weights, n denotes the number of input channels, and $\mathbf{s}$ is the smoothing scale vector.

Despite these developments, existing quantization methods still lead to substantial performance degradation when applied to demoiréing models. Recent methods are mainly designed for LLMs or DiTs and do not adequately address outliers in convolutional modules. Moreover, generic quantization approaches lack task-specific optimization tailored to demoiréing. Therefore, we propose the outlier-aware quantizer and the frequency-aware calibration strategy to enhance the performance.

3.2 Outlier-aware quantizer

Challenges. Despite the progress in neural network quantization, existing methods still face critical limitations, particularly due to the presence of outliers. They can lead to two key problems: ***quantization range expansion*** and ***inaccurate smoothing scale factor estimation***.

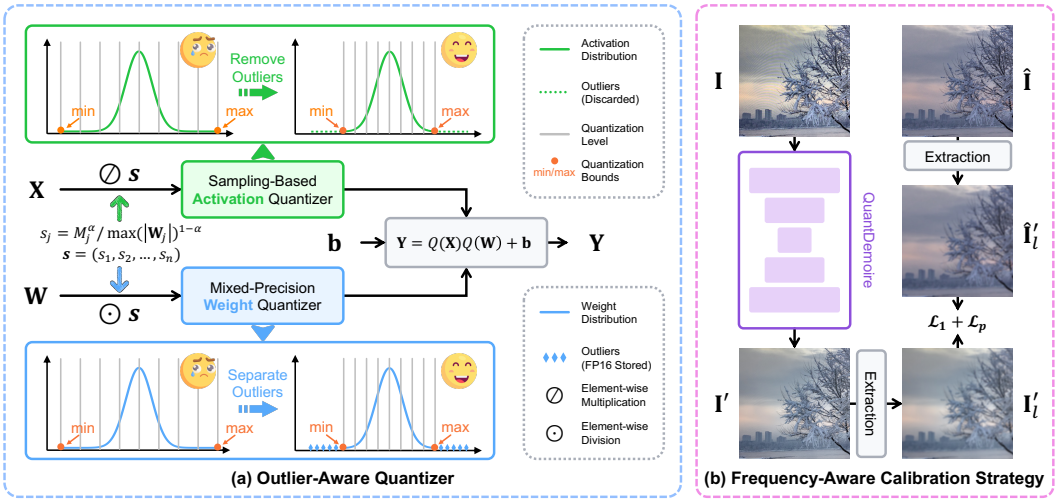

Figure 2: The overview of our QuantDemoire. (a) Outlier-Aware Quantizer: activations discard outliers through sample-based estimation, while weights preserve outliers in FP16 precision. (b) Frequency-Aware Calibration Strategy: quantizer parameters are optimized on mid- and low-frequency components, which are extracted through the frequency extraction process.

Challenge I. Quantization range expansion. Outliers enlarge the quantization range either within a single channel or across the entire tensor, thereby reducing information density of normal values and aggravating quantization error for them, especially under low-bit quantization.

Challenge II. Inaccurate smoothing factor estimation. When the maximum absolute value, which is an outlier, is used to compute the smoothing factor, the maximum values across channels can be aligned. However, as shown in Fig. 3 (Smooth (Raw)), significant range disparities remain among the non-outlier values across channels due to the large difference between outliers and non-outliers.

Some existing PTQ methods [14, 25] are specifically designed to address the problems caused by outliers. However, they are not designed for the outlier distributions of demoiréing models. Consequently, they fail to mitigate quantization errors.

Sampling-Based Activation Quantizer. To address the problems caused by outliers, we propose a random sampling-based method for estimating activation ranges. From Fig. 4, most activations conform to either a Gaussian or an exponential distribution. Such distributions typically exhibit a small number of outliers that deviate from the normal range on one or both sides. Appendix F provides additional activation and weight distribution visualizations to support this point. Furthermore, we validate this distributional observation with Q-Q analysis in Appendix F.1.

To mitigate the impact of these outliers on accuracy, we adopt a strategy of randomly sampling a small subset of activation values in a channel or tensor (Fig. 2a). Sample$_\gamma(\cdot)$ denotes a subset formed by randomly selecting a proportion γ of elements from the activations. This approach reduces the probability of selecting outliers, thereby reducing the quantization error. It also ensures an accurate estimation of the distribution range of the majority of non-outlier values.

In LLMs, outliers arise from the large magnitudes of certain activation channels [31]. This allows the smoothing factor to be calculated directly from the per-channel maximum absolute value. In contrast, demoiréing models, which are primarily composed of convolutional blocks, feature outliers that are mostly sporadic elements. Consequently, the presence of these elemental outliers within a channel can skew the estimation of its overall magnitude. Therefore, we estimate the maximum magnitude of activations for each channel by randomly sampling:

$$M_j = \max(|\text{Sample}_{\gamma_1}(\mathbf{X}_j)|), \tag{6}$$

where $\mathbf{X}_j$ is the j-th input channel of activation, M_j is the maximum estimated, and γ_1 is the proportion of elements sampled from each activation channel. By restricting the estimation to a sampled subset, we effectively capture the main body of activations while reducing the influence of outlier values. For the calculation of the maximum weight magnitude, we directly take the largest absolute value. The computation of the smoothing factor can be written as:

$$s_j = M_j^\alpha / \max(|\mathbf{W}_j|)^{1-\alpha}. \tag{7}$$

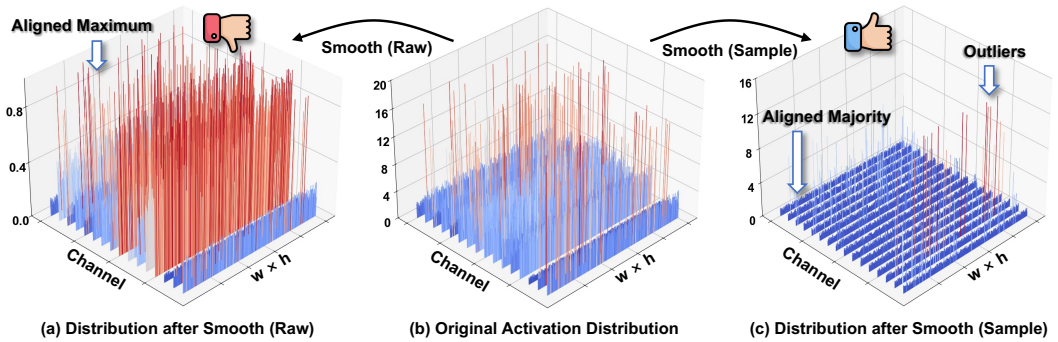

Figure 3: Activation distributions (Original, Smooth (Raw), Smooth (Sample)). (a) Original: ranges differ across channels; (b) Smooth (Raw): align maxima only; (c) Smooth (Sample): align the main body (excluding outliers), improving suitability for quantization.

Visualizations in Fig. 3 reveal that calculating smoothing factor with sampled maximum magnitude of activations (Smooth (Sample)) achieves better smoothing of non-outlier values (*i.e.*, the majority of the distribution). This allocates more quantization levels to ordinary activations. After being divided by the smoothing factor, the activation's per-channel magnitude have been aligned. However, the outlier elements still remain in the activation tensor. These outliers barely improve output quality but greatly increase quantization error, so they should be clipped during quantization. Appendix H.2.3 provides a sensitivity study showing that clipping a tiny fraction of activation outliers changes the convolution output by less than 0.01%, supporting their limited contribution to reconstruction. Therefore, in computing the quantization bound, we still employ a sampling approach, with a proportion γ_2:

$$\begin{aligned} u_a &= \max(\text{Sample}_{\gamma_2}(\mathbf{X} \oslash \mathbf{s})), \\ l_a &= \min(\text{Sample}_{\gamma_2}(\mathbf{X} \oslash \mathbf{s})), \end{aligned} \tag{8}$$

where l_a, u_a denote the lower and upper bounds of the quantizer. $\mathbf{X}$ is the activation tensor and $\mathbf{s}$ is the smoothing factor. Experiments show that computing the quantization bound via random sampling outperforms clipping activations at a fixed percentile in reducing quantization error for demoiréing models. This makes the bound adaptive to tensor size. A mathematical analysis in Appendix H.1 shows the equivalence between sampling and dynamic percentile clipping, and explains why sampling better matches activation outlier distributions across input sizes.

Finally, the derived l_a and u_a are applied in Eq. (1) to perform activation quantization, forming the sampling-based activation quantizer. This quantizer effectively reduces the impact of outliers on quantization accuracy, and improves model performance.

To avoid extra inference-time overhead, the outlier-aware quantizer uses static quantization. We compute the smoothing factor and quantization bounds using a calibration set of 200 training dataset pairs. The calibrated parameters are then reused for all test images. In Appendix H.2.1, we report the variance over multiple runs, showing that our stochastic sampling procedure remains stable.

Mixed-Precision Weight Quantizer. As illustrated in Fig. 4, similar to the behavior observed in activation distributions, the weight distribution of the network exhibits an approximately Gaussian distribution. Therefore, outliers of the weight exist on the left and right side of the distribution. A straightforward way to handle these extreme outlier values is to apply the same sampling-based estimation method proposed in activation quantization. However, unlike activations, weight tensors of convolutional modules generally contain far fewer elements. Consequently, the direct truncation of extreme weight values leads to a notable degradation in the performance of the quantized model. This indicates activation and weight outliers should be treated differently: activation outliers are suppressed through range estimation, whereas weight outliers need to preserve their numerical contribution.

One solution is to store outliers in FP16 via mixed precision. However, most existing quantization methods apply mixed precision at the layer or channel granularity [7]. Because outliers in convolutional-module weights occur sporadically at random positions, these coarse-grained schemes are not directly applicable. They would keep many normal weights in FP16 unnecessarily.

To overcome this limitation, we propose preserving the outliers in FP16 precision. The illustration is in Fig. 2a. By preserving a proportion β of outliers at both ends, we can calculate the corresponding lower and upper percentile thresholds: $T_{\text{low}} = P_\beta(\mathbf{W})$ and $T_{\text{high}} = P_{1-\beta}(\mathbf{W})$. Here, $P_\beta(\mathbf{W})$

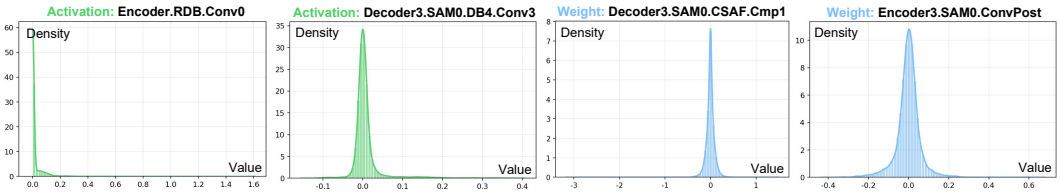

Figure 4: Visualization of **activation** and **weight** distributions from randomly selected layers in ESDNet [36]. The distributions are approximately Gaussian or exponential. Under these distributions, outliers are distributed on both sides. These sparse extremes can dominate the quantization range if they are treated together with ordinary values.

denotes the β-th percentile of $\mathbf{W}$. The complete weight quantization process can then be expressed:

$$\begin{aligned} \mathbf{W}_{\text{outlier}} &= \{w_i \in \mathbf{W} \mid w_i < T_{\text{low}} \ \text{ or } \ w_i > T_{\text{high}}\}, \\ \mathbf{W}_{\text{normal}} &= \mathbf{W} \setminus \mathbf{W}_{\text{outlier}}, \quad \hat{\mathbf{W}} = \mathbf{W}_{\text{outlier}} \cup Q(\mathbf{W}_{\text{normal}}), \end{aligned} \tag{9}$$

where $\mathbf{W}_{\text{outlier}}$ and $\mathbf{W}_{\text{normal}}$ are the outlier and non-outlier weights; $\hat{\mathbf{W}}$ is the final quantized weight. Experimental results in Tab. 2 indicate that $\beta = 0.005$ is sufficient.

During inference, we decompose the convolution into two branches: a low-bit branch and an FP16 branch. The low-bit branch performs the convolution using quantized weights and quantized activations. The FP16 branch computes, on an element-wise basis, the contributions of stored weight outliers to the output. The results of the two branches are then summed. Thus, most computation remains low-bit. Because only a very small fraction of weight outliers are stored in FP16, under W4A4, the FP16 branch introduces only a 6% memory and a 7% runtime overhead.

Conclusion. The outlier-aware quantizer reduces errors caused by outliers in activations and weights. It uses random sampling to estimate the activation range and selectively retain weight outliers in FP16, reducing quantization errors with little overhead. This dual strategy balances efficiency with accuracy. Appendix G provides pseudocode for the sampling-based activation quantizer.

3.3 Frequency-aware calibration strategy

Optimizing quantization boundaries through training can improve the performance to some extent. However, the quantized model still exhibits performance degradation. In particular, quantization affects frequency components unevenly. For high-frequency structures such as edges and textures, the distortions are relatively minor due to their sharp RGB variations. However, in low- and mid-frequency regions with gradual intensity transitions, quantization severely reduces representational capacity, often manifesting as color banding. Appendix H.3 visualizes this banding effect and the improvement brought by frequency-aware calibration.

To address the above problem, we propose a frequency-aware calibration strategy. This method optimizes the parameters of the quantizer by extracting the mid- and low-frequency components from both the model outputs and the ground truth, then optimizing the loss accordingly. A standard spatial-domain loss can be dominated by edges and textures, making it less effective at penalizing low-amplitude errors in smooth mid- and low-frequency regions. Therefore, FAC explicitly aligns the calibration objective with the frequency bands where banding artifacts are most visible. Specifically, following previous work [18, 28], we employ a dedicated convolution kernel $\boldsymbol{k}$ to extract the mid- and low-frequency components. The kernel is defined as:

$$\boldsymbol{k} = \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}. \tag{10}$$

The frequency extraction process is an L-step iterative procedure based on kernel $\boldsymbol{k}$. Given a image $\mathbf{I}$, in the i-th step (i = 1, 2, ..., L), the extraction can be formalized as:

$$\mathbf{I}^i = E^i_{\boldsymbol{k}}(\mathbf{I}^{i-1}), \tag{11}$$

where $\mathbf{I}^0$=$\mathbf{I}$, $\mathbf{I}^i$ is the outcome of the i-th step of extraction. $E^i_{\boldsymbol{k}}$ denotes an operation where the input is convolved with the kernel $\boldsymbol{k}$ and a dilation of 2^i. The whole **frequency extraction** process can be written as a recursive function $F(\cdot)$, which is defined as follows:

$$F(\mathbf{I}, i) = \begin{cases} \mathbf{I}, & i = 0, \\ E^i_{\boldsymbol{k}}\big(F(\mathbf{I}, i-1)\big), & i \geq 1. \end{cases} \tag{12}$$

Method	MinMax	Percentile	Sample
PSNR ↑	16.51	16.85	**17.61**
SSIM ↑	0.5255	0.6701	**0.6892**
LPIPS ↓	0.6786	0.4639	**0.4552**

(a) Quantizer bound calculation.

Method	Original	Smooth (Raw)	Smooth (Sample)
PSNR ↑	17.61	20.43	**20.52**
SSIM ↑	0.6892	0.7408	**0.7538**
LPIPS ↓	0.4552	0.3363	**0.3247**

(b) Smooth transformation.

Table 1: Ablation study on sampling-based activation quantizer.

Method	Baseline	Discard Outliers	Store Random	Store Outlier
PSNR ↑	20.52	16.29	20.54	**20.92**
SSIM ↑	0.7538	0.6724	0.7542	**0.7570**
LPIPS ↓	0.3247	0.4103	0.3244	**0.3171**

(a) Outlier handling method.

#β	PSNR ↑	SSIM ↑	LPIPS ↓	Params (M)	OPs (G)
0.25%	20.88	0.7565	0.3180	0.7837	1.78
0.50%	20.92	0.7570	0.3171	0.7948	1.80
1.00%	21.04	0.7578	0.3171	0.8171	1.85

(b) Outliers Proportion (β).

Table 2: Ablation study on mixed-precision weight quantizer.

The outcome of the whole frequency extraction procedure is $F(\mathbf{I}, L)$. As Fig. 2b, during the phase of optimizing quantization boundaries, we extract the low- and mid-frequency components from the output image and ground truth. This keeps the calibration target in the image domain while emphasizing smooth regions. Then following ESDNet [36], we directly reuse these losses: a pixel-wise $\mathcal{L}_1$ loss and a feature-based perceptual loss $\mathcal{L}_p$. The perceptual loss $\mathcal{L}_p$ is computed as the L_1 distance between the VGG16 extracted feature [36].

Specifically, given the output of the quantitative model $\mathbf{I}'$ and the corresponding ground truth $\hat{\mathbf{I}}$, the extracted component of low and mid frequencies are defined as $\mathbf{I}'_l = F(\mathbf{I}', L)$ and $\hat{\mathbf{I}}_l = F(\hat{\mathbf{I}}, L)$, respectively. Then, we compute the loss $\mathcal{L}_{\text{total}}$ on the extracted frequencies as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1(\mathbf{I}'_l, \hat{\mathbf{I}}_l) + \mathcal{L}_p(\mathbf{I}'_l, \hat{\mathbf{I}}_l). \tag{13}$$

We only optimize the quantization boundaries of activations while keeping all other parameters fixed. Besides, we find that when L is small (*e.g.*, $L = 1$), more high-frequency information is preserved, whereas a larger L (*e.g.*, $L = 5$) focuses only on low frequencies. Since our goal is to focus on both low- and mid-frequency regions, we set $L = 3$. Experiments in Sec. 4.2 (Tab. 3b) validate this choice. Since the extraction module introduces only a few convolutions, the overhead is negligible. Besides, FAC changes the calibration signal rather than the model, and therefore does not affect inference.

4 Experiments

4.1 Experimental Settings

Datasets and Metrics. We evaluate on three moiré pattern removal datasets: UHDM [36], FHDMi [9], and LCDMoiré [37]. For each dataset, we randomly select 200 training image pairs and crop them to construct a calibration dataset. After the calibration stage, we evaluate the results on the corresponding test dataset. For quantitative assessment, we adopt PSNR, SSIM [30], and LPIPS [38].

Implementation Details. We adopt ESDNet [36] as the backbone and provide results under 3, 4, 6, and 8 bits. The quantization setup is denoted as WwAa, where w and a are the bit widths of weights and activations. For the sampling rate, we set $\gamma_1 = \gamma_2 = 10^{-3}$. For the smoothing factor, we use $\alpha = 0.5$. Both our method and the compared baselines apply static quantization, with per-channel quantization for weights and per-tensor quantization for activations. The baselines and our method are fairly evaluated, with more details provided in Appendix B. For the FP16 weight branch, since there's currently no way to handle FP16 convolution outliers, we implemented a fused CUDA kernel.

Calibration Settings. During the optimization stage, the quantization parameters are trained for 4 epochs using the Adam [12] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The image pairs are randomly cropped into 512×512, and the batch size is 1. We apply the loss proposed in Eq. (13). The initial learning rate is set to 0.001 and scheduled with cyclic cosine annealing [20]. All codes are implemented with PyTorch [22], and all experiments are conducted on one A6000-48G GPU.

4.2 Ablation Study

We investigate the impact of the proposed outlier-aware quantizer (for activations and weights) and the frequency-aware calibration strategy. All experiments are conducted on the UHDM [36] dataset with 4-bit quantization (W4A4). The settings follow Sec. 4.1 to ensure fairness.

Method	No Calibration	Orignal Image	High-Freq	Mid&Low-Freq
PSNR ↑	20.92	20.95	20.62	**21.08**
SSIM ↑	0.7570	0.7604	0.7538	**0.7626**
LPIPS ↓	0.3171	0.3088	0.3275	**0.3068**

(a) Calibration strategy.

Level	1	3	5
PSNR ↑	20.96	**21.08**	21.05
SSIM ↑	0.7605	**0.7626**	0.7605
LPIPS ↓	0.3092	**0.3068**	0.3071

(b) Frequency extraction level.

Table 3: Ablation study on frequency-aware calibration strategy.

Method	Bit	UHDM			FHDMi			LCDMoiré		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
ESDNet [36]	W32A32	22.12	0.7956	0.2551	24.50	0.8351	0.1354	44.83	0.9963	0.0097
MinMax [10]	W8A8	21.50	0.7727	0.2596	20.30	0.7631	0.2600	37.92	0.9865	0.0236
Percentile [14]	W8A8	19.38	0.7744	0.2784	19.73	0.7519	0.2734	30.11	0.9585	0.0281
2DQuant [17]	W8A8	21.20	0.7827	0.2749	20.57	0.7861	0.2034	41.60	0.9923	0.0214
SVDQuant [13]	W8A8	21.80	0.7907	0.2580	22.07	0.7966	0.1598	41.18	0.9915	0.0165
QuantDemoire (ours)	W8A8	22.00	0.7932	0.2555	22.23	0.8026	0.1591	42.16	0.9930	0.0126
MinMax [10]	W6A6	20.48	0.7648	0.2828	19.85	0.7362	0.3040	24.53	0.9337	0.1692
Percentile [14]	W6A6	18.44	0.7562	0.2957	19.22	0.7353	0.2953	22.46	0.9342	0.1074
2DQuant [17]	W6A6	20.02	0.7595	0.2893	20.66	0.7389	0.2558	26.53	0.9366	0.2241
SVDQuant [13]	W6A6	20.86	0.7602	0.3015	20.91	0.6883	0.3732	32.10	0.9691	0.1314
QuantDemoire (ours)	W6A6	21.61	0.7874	0.2572	21.63	0.7861	0.1721	40.48	0.9910	0.0206
MinMax [10]	W4A4	16.51	0.5255	0.6786	16.01	0.4958	0.6686	15.14	0.7031	0.7500
Percentile [14]	W4A4	16.85	0.6701	0.4639	16.92	0.6424	0.4858	14.95	0.8290	0.5596
2DQuant [17]	W4A4	17.07	0.6288	0.5117	14.80	0.3579	0.7107	15.11	0.6075	0.7561
SVDQuant [13]	W4A4	14.68	0.5762	0.6559	12.43	0.2818	0.7961	15.25	0.7201	0.6687
QuantDemoire (ours)	W4A4	21.08	0.7626	0.3068	20.28	0.6745	0.3369	31.28	0.9544	0.1793
MinMax [10]	W3A3	12.53	0.3802	0.8630	12.16	0.2903	0.8514	10.11	0.6051	0.8376
Percentile [14]	W3A3	14.79	0.5206	0.6869	14.33	0.4444	0.7496	9.30	0.5242	1.0272
2DQuant [17]	W3A3	11.20	0.2465	0.8460	10.92	0.2879	0.8612	9.07	0.3026	1.0153
SVDQuant [13]	W3A3	14.83	0.4547	0.7289	9.34	0.2127	0.8840	10.17	0.5658	0.9753
QuantDemoire (ours)	W3A3	19.12	0.6839	0.4567	18.26	0.5722	0.4930	22.02	0.8558	0.4544

Table 4: Quantitative comparison. The best and second-best results are colored red and blue.

Activation Quantizer. Tab. 1a compares activation bound estimation methods. The baseline is the min-max method [10]. We compare it with the percentile [14] and our sample-based method (as in Eq. (7)). Compared with previous boundary selection, our sampling-based approach is more effective.

Furthermore, we introduce a smooth transformation to adjust channels within the sample-based framework. As illustrated in Fig. 3, we compare two variants: the original definition and our sample-based smoothing. Results in Tab. 1b show that outlier removal via sampling enhances the effectiveness of the smoothing. In general, our sample-based activation quantizer leads to more robust quantization.

Weight Quantizer. We compare different strategies for handling weight outliers in Tab. 2a. We adopt the smooth (sample-based) result from Tab. 1b as the baseline. The results show that, compared with randomly storing some weights in FP16, explicitly preserving outliers in FP16 is more effective. Besides, consistent with the analysis in Sec. 3.2, discarding outliers leads to a performance drop.

We further evaluate the proportion of outliers (β) in Tab. 2b. The OPs are computed under an input size of $3\times224\times224$. We find that setting the ratio β=0.5% achieves a favorable trade-off between efficiency and performance. Therefore, we adopt this setting as the default in QuantDemoire.

Calibration Strategy. We conduct an ablation study on different calibration strategies for optimizing quantized parameters. We apply the outlier-aware quantizer as the baseline. Results in Tab. 3a show that focusing on mid- and low-frequency components yields better performance.

We also evaluate frequency extraction levels (Tab. 3b). The relatively high (level=1) or low (level=5) frequencies are not optimal. In contrast, level=3 achieves better performance, which aligns with our analysis. Besides, traditional image quality assessment (IQA) metrics do not accurately reflect the impact of banding artifacts [29]. Therefore, Appendix H.3 presents additional visual results comparing the original spatial-domain loss with frequency-aware calibration. Appendix A and Appendix H.5 provide additional sampling-rate ablations and the full settings for all ablation variants.

4.3 Comparison with State-of-the-Art Methods

We utilize ESDNet [36] as the full-precision backbone and compare QuantDemoire with several advanced post-training quantization methods, including MinMax [10], Percentile [14], 2DQuant [17], and SVDQuant [13]. Moreover, Appendix C further analyzes SVDQuant with different ranks, Appendix H.2.5 evaluates QuantDemoire on another demoiréing backbone and an image super-resolution model, and Appendix H.4 reports end-to-end latency.

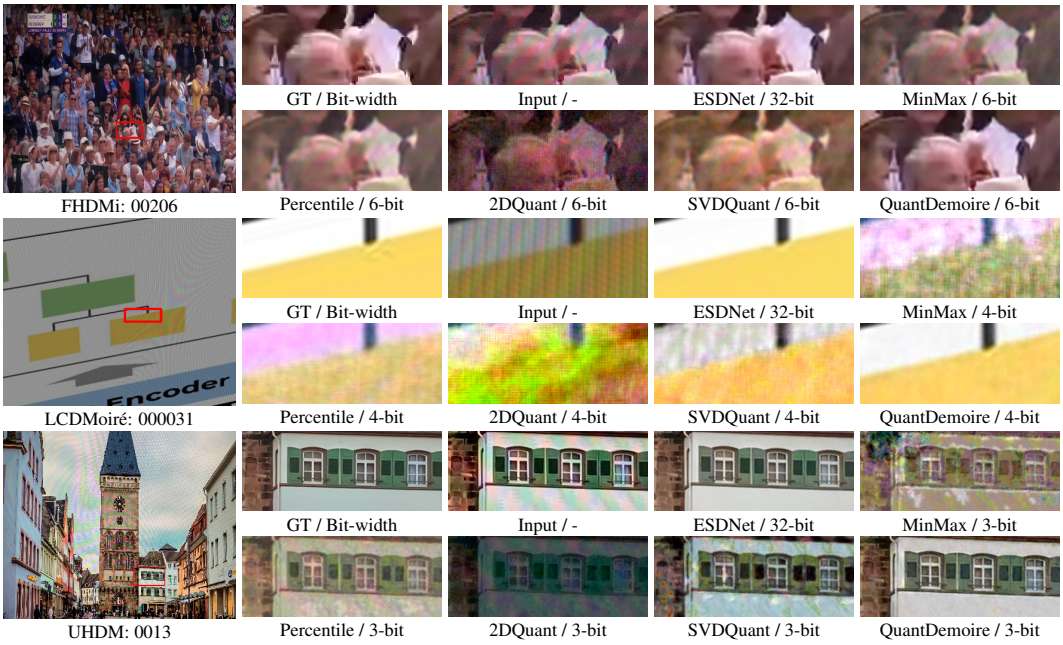

Figure 5: Visual comparison on UHDM [36], FHDMi [9], and LCDMoiré [37] datasets.

Quantitative Results. Quantitative evaluations are summarized in Tab. 4. Across all datasets, bit-widths, and metrics, our method outperforms prior quantization methods, with larger gains at low bit-widths (*e.g.*, 3/4-bit). For example, in the 4-bit setting, our approach surpasses the latest method, SVDQuant [13], by more than **6 dB** on the UHDM [36] dataset. Furthermore, our proposed method retains performance close to the full-precision network under 8/6-bit settings, whereas the degradation at 3/4-bit is greatly reduced. These results demonstrate the effectiveness of our approach.

Qualitative Results. Fig. 5 shows 3/4/6-bit comparisons. Existing quantization baselines tend to leave residual moiré artifacts or fail to reconstruct clean structures, especially on difficult samples and at lower precision. Our QuantDemoire removes moiré patterns and restores fine details, producing results close to full precision. This observation is consistent with the quantitative results in Tab. 4. Appendix E provides additional visual comparisons across datasets and bit-widths.

Compression Ratio. We report the parameter (Params) and operation (OPs) compression ratios at 6/4/3-bit in Tab. 5. The OPs are measured with an input size of $3\times224\times224$. Compared with the full-precision (32-bit) model, our approach significantly reduces both model size and computational cost. At 4-bit, the compression rate exceeds 86.6%, while the performance drop is limited to 4.7% (PSNR). Moreover, compared with existing methods, QuantDemoire demonstrates clear advantages. Appendix D provides a more comprehensive compression-ratio comparison.

Method	Bit (w/a)	Params (M) (↓Ratio)	OPs (G) (↓Ratio)	UHDM PSNR ↑	UHDM SSIM ↑
ESDNet	32/32	5.93 (↓0%)	13.52 (↓0%)	22.12	0.7956
2DQuant	6/6	1.12 (↓81.14%)	2.54 (↓81.25%)	20.02	0.7595
SVDQuant	6/6	1.45 (↓75.53%)	4.60 (↓65.97%)	20.86	0.7602
QuantDemoire	6/6	1.16 (↓80.43%)	2.64 (↓80.49%)	21.61	0.7874
2DQuant	4/4	0.75 (↓87.38%)	1.69 (↓87.50%)	17.07	0.6288
SVDQuant	4/4	1.08 (↓81.78%)	3.76 (↓72.19%)	14.68	0.5762
QuantDemoire	4/4	0.79 (↓86.61%)	1.80 (↓86.68%)	21.08	0.7626
2DQuant	3/3	0.56 (↓90.51%)	1.27 (↓90.63%)	11.20	0.2465
SVDQuant	3/3	0.90 (↓84.89%)	3.34 (↓75.30%)	14.83	0.4547
QuantDemoire	3/3	0.61 (↓89.69%)	1.38 (↓89.78%)	19.12	0.6839

Table 5: Compression ratios of Params and OPs at 6/4/3-bit. OPs are measured with an input size of $3\times224\times224$.

5 Conclusion

In this paper, we propose QuantDemoire, a post-training quantization method tailored for image demoiréing. Our design is developed from two perspectives: the quantizer and the calibration strategy. We first introduce the outlier-aware quantizer to reduce quantization errors caused by outliers. We Besides, we develop a frequency-aware calibration strategy emphasizing mid- and low-frequency components to mitigate banding artifacts caused by low-bit quantization. Comprehensive experiments verify that our method effectively reduces overhead compared with the full-precision model. Meanwhile, QuantDemoire outperforms current advanced quantization methods.

Acknowledgments and Disclosure of Funding

This work was supported by Huawei Explore X funds. This work was also sponsored by AI Hundred Schools Program and was carried out using the Ascend AI technology stack.

Competing interests. The authors declare no additional competing interests beyond the affiliations and funding disclosed above.

References

[1] Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*, 2024.

[2] Xi Cheng, Zhenyong Fu, and Jian Yang. Multi-scale dynamic feature encoding network for image demoiréing. In *ICCVW*, 2019.

[3] Yongsen Cheng, Kai Liu, Kaiwen Tao, Junxian Li, Zhixin Wang, Zhikai Chen, Renjing Pei, and Yulun Zhang. Permuquant: Lowering per-group quantization error by reordering channels for diffusion models. *arXiv preprint arXiv:2605.09503*, 2026.

[4] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. In *NeurIPS*, 2022.

[5] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35:30318–30332, 2022.

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

[7] Weilun Feng, Haotong Qin, Chuanguang Yang, Zhulin An, Libo Huang, Boyu Diao, Fei Wang, Renshuai Tao, Yongjun Xu, and Michele Magno. Mpq-dm: Mixed precision quantization for extremely low bit diffusion models. In *AAAI*, 2025.

[8] Bin He, Ce Wang, Boxin Shi, and Ling-Yu Duan. Mop moire patterns using mopnet. In *ICCV*, 2019.

[9] Bin He, Ce Wang, Boxin Shi, and Ling-Yu Duan. Fhde2net: Full high definition demoireing network. In *ECCV*, 2020.

[10] 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 *CVPR*, 2018.

[11] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*, 2023.

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[13] 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.

[14] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *CVPR*, 2019.

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

[16] Fanglei Liu, Jingyu Yang, and Huanjing Yue. Moiré pattern removal from texture images via low-rank and sparse matrix decomposition. In *VCIP*, 2015.

[17] 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. In *NeurIPS*, 2024.

[18] Xiaoyang Liu, Bolin Qiu, Jiezhang Cao, Zheng Chen, Yulun Zhang, and Xiaokang Yang. Freqformer: Image-demoir\'eing transformer via efficient frequency decomposition. *arXiv preprint arXiv:2505.19120*, 2025.

[19] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. In *NeurIPS*, 2021.

[20] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[21] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[23] Haotong Qin, Yulun Zhang, Yifu Ding, Xianglong Liu, Martin Danelljan, Fisher Yu, et al. Quantsr: accurate low-bit quantization for efficient image super-resolution. In *NeurIPS*, 2023.

[24] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *CVPR*, 2023.

[25] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*, 2023.

[26] Denis N Sidorov and Anil Christopher Kokaram. Suppression of moiré patterns via spectral analysis. In *VCIP*, 2002.

[27] Ce Wang, Bin He, Shengsen Wu, Renjie Wan, Boxin Shi, and Ling-Yu Duan. Coarse-to-fine disentangling demoiréing framework for recaptured screen images. *TPAMI*, 2023.

[28] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *IJCV*, 2024.

[29] Yilin Wang, Sang-Uok Kum, Chao Chen, and Anil Kokaram. A perceptual visibility metric for banding artifacts. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2067–2071. IEEE, 2016.

[30] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

[31] 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 *ICML*, 2023.

[32] Zeyu Xiao, Zhihe Lu, and Xinchao Wang. P-bic: Ultra-high-definition image moiré patterns removal via patch bilateral compensation. In *ACM MM*, 2024.

[33] Shuning Xu, Binbin Song, Xiangyu Chen, and Jiantao Zhou. Image demoireing in raw and srgb domains. *arXiv preprint arXiv:2312.09063*, 2023.

[34] Xianglong Yan, Chengzhu Bao, Zhiteng Li, Tianao Zhang, Kaicheng Yang, Haotong Qin, Ruobing Xie, Samm Sun, and Yulun Zhang. Pt^2-llm: Post-training ternarization for large language models. In *ICLR*, 2026.

[35] Xianglong Yan, Hong Liu, Chengzhu Bao, Tianao Zhang, Guanghua Yu, Jianchen Zhu, and Yulun Zhang. Focus: Fp4 optimization via coupled-relaxation and dual-granularity scaling. *arXiv preprint arXiv:2608.01847*, 2026.

[36] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Jiajun Shen, Jia Li, and Xiaojuan Qi. Towards efficient and scale-robust ultra-high-definition image demoiréing. In *ECCV*, 2022.

[37] Shanxin Yuan, Radu Timofte, Gregory Slabaugh, Aleš Leonardis, Bolun Zheng, Xin Ye, Xiang Tian, Yaowu Chen, Xi Cheng, Zhenyong Fu, et al. Aim 2019 challenge on image demoireing: Methods and results. In *ICCVW*, 2019.

[38] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[39] Bolun Zheng, Shanxin Yuan, Gregory Slabaugh, and Ales Leonardis. Image demoireing with learnable bandpass filters. In *CVPR*, 2020.

[40] Zihan Zhou, Libo Zhu, Jue Gong, Zhiyi Zhou, Jiezhang Cao, Yong Guo, and Yulun Zhang. Bricker to BRACE: A bracket exposure raw dataset and restoration model for flicker-banding. In *NeurIPS*, 2026.

[41] Libo Zhu, Jianze Li, Haotong Qin, Yulun Zhang, Yong Guo, and Xiaokang Yang. PassionSR: Post-training quantization with adaptive scale in one-step diffusion based image super-resolution. In *CVPR*, 2025.

A More Ablation

Rate	10^{-2}	10^{-3}	10^{-4}
PSNR ↑	19.92	**20.92**	20.80
SSIM ↑	0.6941	**0.7570**	0.7505
LPIPS ↓	0.4141	**0.3171**	0.3221

(a) Smooth sampling rate (γ_1).

Rate	10^{-2}	10^{-3}	10^{-4}
PSNR ↑	20.88	**20.92**	20.89
SSIM ↑	0.7558	**0.7570**	0.7474
LPIPS ↓	0.3183	**0.3171**	0.3388

(b) Bound sampling rate (γ_2).

Table 6: Ablation on the sampling rate in the activation. Evaluation is conducted on UDHM, 4-bit.

For the smooth stage (γ_1), we ablate the sampling rate by varying γ_1 while fixing $\gamma_2 = 10^{-3}$. As shown in Tab. 6a, the setting that produces the best performance corresponds to $\gamma_1 = 10^{-3}$. Higher rates introduce excessive outliers in the smoothing process, whereas lower rates bias the preliminary estimate. A moderate sampling rate, therefore, provides the most stable and accurate range estimation.

For the bound stage (γ_2), we very γ_2 while fixing $\gamma_1 = 10^{-3}$. Tab. 6b indicates that $\gamma_2 = 10^{-3}$ gives the best results. Larger rates lead to the accumulation of outliers during refinement, and smaller rates cause biased bounds. These results confirm that a moderate sampling rate also optimizes accuracy, reinforcing our overall design choice.

B Details of Comparison Methods

MinMax. We apply per-channel quantization to the weights and per-tensor quantization to the activations. During the calibration, we directly use the minimum and maximum activation values as the quantization boundaries.

Percentile. We also use per-channel quantization for the weights and per-tensor quantization for the activations. Specifically, for both weights and activation values, the 99.9%-th and 0.1%-th percentiles of all values are consistently used as the quantization boundaries.

2DQuant. We first employ the DOBI [17] method to search for quantizer parameters. Then we train the model using the quantization parameters obtained from DQC [17], with a training configuration of 100 epochs, a batch size of 4, and a learning rate of 10^{-2} . All other settings remain consistent with our proposed method.

SVDQuant. We first apply SmoothQuant [31] to reduce activations outliers. Next, we perform SVD on the weights and compute both the low-rank and residual branches. To ensure fairness in our experiments and to prevent the low-rank branch from introducing an excessive number of additional parameters, we set the rank to 2.

C Variants of SVDQuant

We observed that SVDQuant suffers from a substantial performance drop when quantized to extremely low bit-widths (3 and 4 bits). To investigate this issue, we conducted further experiments with higher ranks. The results in Tab. 7 show that when the rank is increased to 8, SVDQuant achieves a significant performance improvement. However, when the rank is set to 8, although the performance improves, the relatively small parameter size of the convolutional layers in ESDNet leads SVDQuant to introduce an excessive number of additional parameters. Therefore, in our experiments, we set rank to 2 for comparative fairness.

D More Compression Ratio

A more comprehensive comparison of compression ratios is in Tab. 8. Our method achieves performance close to FP32 model, while maintaining a high compression ratio.

E More Qualitative Results

Additional visual comparisons are presented in Figs. 8 and 9. Our method demonstrates clear advantages under the 3-, 4-, and 6-bit settings, as well as across all datasets.

Method	Bit(w/a)	Params (M)	OPs (G)	UHDM PSNR ↑	UHDM SSIM ↑	UHDM LPIPS ↓
SVDQuant (rank = 2)	4/4	1.08	3.76	14.68	0.5762	0.6559
SVDQuant (rank = 8)	4/4	2.10	8.48	18.69	0.7428	0.3615
QuantDemoire (ours)	4/4	**0.79**	**1.80**	**21.08**	**0.7626**	**0.3068**
SVDQuant (rank = 2)	3/3	0.90	3.34	14.83	0.4547	0.7289
SVDQuant (rank = 8)	3/3	2.01	6.80	17.20	0.6707	0.5033
QuantDemoire (ours)	3/3	**0.61**	**1.38**	**19.12**	**0.6839**	**0.4567**

Table 7: Results of SVDQuant (rank = 2 and 8). Evaluation is conducted on UDHM, 3 and 4-bit.

Method	Bit (w/a)	Params (M) (↓Ratio)	OPs (G) (↓Ratio)	UHDM PSNR ↑	UHDM SSIM ↑	UHDM LPIPS ↓
ESDNet [36]	32/32	5.93 (↓0%)	13.52 (↓0%)	22.12	0.7956	0.2551
MinMax [10]	8/8	1.49 (↓74.90%)	3.38 (↓75.00%)	21.50	0.7727	0.2596
Percentile [14]	8/8	1.49 (↓74.90%)	3.38 (↓75.00%)	19.38	0.7744	0.2784
2DQuant [17]	8/8	1.49 (↓74.90%)	3.38 (↓75.00%)	21.20	0.7827	0.2749
SVDQuant [13]	8/8	1.82 (↓69.29%)	5.44 (↓59.75%)	21.80	0.7907	0.2580
QuantDemoire (ours)	8/8	1.53 (↓74.25%)	3.47 (↓74.31%)	22.00	0.7932	0.2555
MinMax [10]	6/6	1.12 (↓81.14%)	2.54 (↓81.25%)	20.48	0.7648	0.2828
Percentile [14]	6/6	1.12 (↓81.14%)	2.54 (↓81.25%)	18.44	0.7562	0.2957
2DQuant [17]	6/6	1.12 (↓81.14%)	2.54 (↓81.25%)	20.02	0.7595	0.2893
SVDQuant [13]	6/6	1.45 (↓75.53%)	4.60 (↓65.97%)	20.86	0.7602	0.3015
QuantDemoire (ours)	6/6	1.16 (↓80.43%)	2.64 (↓80.49%)	21.61	0.7874	0.2572
MinMax [10]	4/4	0.75 (↓87.38%)	1.69 (↓87.50%)	16.51	0.5255	0.6786
Percentile [14]	4/4	0.75 (↓87.38%)	1.69 (↓87.50%)	16.85	0.6701	0.4639
2DQuant [17]	4/4	0.75 (↓87.38%)	1.69 (↓87.50%)	17.07	0.6288	0.5117
SVDQuant [13]	4/4	1.08 (↓81.78%)	3.76 (↓72.19%)	14.68	0.5762	0.6559
QuantDemoire (ours)	4/4	0.79 (↓86.61%)	1.80 (↓86.68%)	21.08	0.7626	0.3068
MinMax [10]	3/3	0.56 (↓90.51%)	1.27 (↓90.63%)	12.53	0.3802	0.8630
Percentile [14]	3/3	0.56 (↓90.51%)	1.27 (↓90.63%)	14.79	0.5206	0.6869
2DQuant [17]	3/3	0.56 (↓90.51%)	1.27 (↓90.63%)	11.20	0.2465	0.8460
SVDQuant [13]	3/3	0.90 (↓84.89%)	3.34 (↓75.30%)	14.83	0.4547	0.7289
QuantDemoire (ours)	3/3	0.61 (↓89.69%)	1.38 (↓89.78%)	19.12	0.6839	0.4567

Table 8: Compression ratios of Params and OPs. OPs are measured with a $3\times224\times224$ input.

F More Distribution Visualizations

Additional distributions of weights and activations are presented in Figs. 10, 11, 12, and 13. The activations in most convolutional layers approximately follow either an exponential or a Gaussian distribution, while the weights in the majority of convolutional layers exhibit an approximately Gaussian distribution overall. This further proves our point.

F.1 Q-Q Analysis of Activation Distributions

To further validate the distributional observation, we perform a Gaussian Q-Q analysis on the activation tensor of the layer `decoder.decoder_2.sam_blocks.0.fusion.compress1`. The Q-Q correlation reaches 0.9916, indicating that the main body of the activation distribution is close to Gaussian. Importantly, QuantDemoire does not depend on an exact Gaussian assumption; the key property used by our sampling estimator is the sparse occurrence of extreme outliers.

G Pseudocode of QuantDemoire

To provide a clearer description of the procedure of our method, we present the following pseudocode in Alg. 1 for the sampling-based activation quantizer.

Algorithm 1 Pipeline of Sampling-Based Activation Quantizer

Initialization:
$model \leftarrow \text{ESDNet(fp32)}$
$N \leftarrow$ number of batches in calibration dataset
for convolution layers L_i in $model$ **do**
  $W_i \leftarrow$ pretrained weight of L_i
  $C_i \leftarrow$ number of input channels of L_i
  $lb_a^i \leftarrow 0,\ ub_a^i \leftarrow 0$
  $lb_w^i \leftarrow 0,\ ub_w^i \leftarrow 0$
  $\boldsymbol{s}^i[1:C_i] \leftarrow 0$
end for
Calibration Stage 1:
for x in calibration dataset **do**
  $model(x)$
  for convolution layers L_i in $model$ **do**
    $\mathbf{A} \leftarrow$ latest activation tensor of L_i
    $\boldsymbol{s}^i[c] \leftarrow \boldsymbol{s}^i[c] + \max\left|\text{Sample}_\gamma(\mathbf{A}_c)\right|,\ \forall c = 1, \ldots, C_i$
  end for
end for
for convolution layers L_i in $model$ **do**
  $\boldsymbol{s}^i \leftarrow \boldsymbol{s}^i / N$
end for
Calibration Stage 2:
for x in calibration dataset **do**
  $model(x)$
  for convolution layers L_i in $model$ **do**
    $\mathbf{A} \leftarrow$ latest activation tensor of L_i
    $lb_a^i \leftarrow lb_a^i + \min(\text{Sample}_\gamma(\mathbf{A} \oslash \boldsymbol{s}^i))$
    $ub_a^i \leftarrow ub_a^i + \max(\text{Sample}_\gamma(\mathbf{A} \oslash \boldsymbol{s}^i))$
  end for
end for
for convolution layers L_i in $model$ **do**
  $lb_a^i \leftarrow lb_a^i / N$
  $ub_a^i \leftarrow ub_a^i / N$
  $W_i \leftarrow W_i \oslash \boldsymbol{s}^i$
end for

H More Details

H.1 Theoretical Analysis of Sampling Approach

Based on the following analysis, we will demonstrate that: (1) under the activation distribution of the demoiréing model, the sampling-based estimation is equivalent to percentile clipping with a dynamically varying percentile that adapts to the number of activations, and this dynamic adjustment better aligns with the actual distribution characteristics of outliers; (2) the variance of the equivalent percentile is small, thereby demonstrating the stability of our method.

H.1.1 Equivalence Between Sampling and Dynamic Percentile Clipping

As illustrated in Fig. 10 and Fig. 11, the activation distributions of nearly all convolutional layers approximate a normal distribution following the smoothing operation. Consequently, we model elements N within a given activation tensor as N independent and identically distributed random variables (i.i.d.) drawn from a standard normal distribution. We used a sampling ratio of $\gamma = 0.1\%$, resulting in a sample size of $n = \gamma N$. Formally, let $X_1, \ldots, X_N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ and denote the maximum of all values by $M_N = \max_{1 \le i \le N} X_i$. Its probability density function (PDF):

$$f_{M_N}(x) = N\phi(x)\Phi(x)^{N-1}, \tag{14}$$

N	$\|x - mean\| > 4 * std$	$\|x - mean\| > 4.5 * std$
10^6	51	17
10^7	103	41
10^8	253	72

Table 9: Results of number of extreme outliers $|x - mean| > 4 * std$ means the number of activations that lies more than **four** standard deviations away from the mean.

where ϕ and Φ are the standard normal PDF and cumulative distribution function (CDF). The expectation is then:

$$\mathbb{E}[M_N] = \int_{-\infty}^{\infty} xN\phi(x)\Phi(x)^{N-1}dx. \tag{15}$$

Similarly, we can compute the expected maximum among n activation values selected at random as follows:

$$\mathbb{E}[M_n] = \int_{-\infty}^{\infty} xn\phi(x)\Phi(x)^{n-1}dx. \tag{16}$$

This allows us to treat the sampling-based estimation as equivalent to clipping at a percentile threshold, p, for our sampling-based estimation using the CDF: $p = \Phi(\mathbb{E}[M_n])$. We observe that as the total number of elements (N) increases, the sampling-based estimation corresponds to a higher equivalent percentile threshold. For example, for $N = 10^6$, $p \approx 99.95\%$; for $N = 10^7$, $p \approx 99.995\%$; and for $N = 10^8$, $p \approx 99.9995\%$. Furthermore, we find that due to the smoothing operation, the number of extreme outliers tends to stabilize for very large N ($N > 10^6$) rather than growing linearly, as shown in Tab 9.

Therefore, by dynamically clipping at a higher percentile for larger N, our method effectively removes extreme outliers while avoiding incorrectly truncate normal values.

H.1.2 Variance Analysis

Based on this analysis, we calculate the standard deviation of p. Using a first-order Taylor expansion (the Delta Method), the standard deviation can be approximated as:

$$\begin{aligned} \mathbb{E}[M_n^2] &= \int_{-\infty}^{\infty} x^2 f_{M_n}(x)dx \\ &= \int_{-\infty}^{\infty} x^2 n\phi(x)\Phi(x)^{n-1}dx, \end{aligned} \tag{17}$$

Numerical calculations for $N = 10^6, 10^7$and 10^8 yield standard deviations of 6.522×10^{-4}, 6.596×10^{-5}, and 6.653×10^{-6}, respectively, which are negligible.

H.2 Additional Experimental Results

H.2.1 Stability and Reproducibility

To validate the stability of our method, we assessed the variance by conducting five runs as shown in Tab. 10.

The standard deviation of each metric across five runs is negligible, demonstrating the stability of our method.

H.2.2 More Comparison with fixed percentile

We present additional experiments demonstrating that our sampling method outperforms fixed-percentile clipping. Specifically, we compare sampling with 95th- and 99.9th-percentile clipping. The results in Tab. 11 show that sampling provides clear advantages over fixed-percentile.

RUN	PSNR ↑	SSIM ↑	LPIPS ↓
1	21.0808	0.7626	0.3068
2	21.0835	0.7623	0.3066
3	21.0798	0.7628	0.3070
4	21.0801	0.7624	0.3059
5	21.0824	0.7631	0.3062
std	0.0014	0.0003	0.0004

Table 10: Results of PSNR, SSIM, and LPIPS metrics across 5 runs.

Method	PSNR	SSIM	LPIPS
Ours (sampling)	20.2803	0.6745	0.3369
percentile 95%	19.3058	0.6026	0.3623
percentile 99.9%	19.3526	0.6091	0.3487

Table 11: Comparison of different methods. Apart from using sampling-based estimation, % clipping, and 99.9% clipping to compute the quantization boundaries and the smoothing factor. The remaining settings for all three methods are identical to the final configuration of our method.

H.2.3 Sensitivity to Activation Outlier Clipping

We evaluate the effect of clipping rare activation outliers by comparing the convolution outputs before and after clipping. Even when clipping activations at the 99.9th percentile, the relative output error remains below 0.01%, suggesting that these extreme activation values contribute little to reconstruction but substantially enlarge the quantization range.

H.2.4 Initialization vs. Optimization

Because rounding in quantization can destabilize training, training alone can only reduce quantization error within a small range; a well-chosen initialization of the quantization boundary remains crucial. The sampled-activation quantizer approach aims to initialize the quantization boundary to a favorable starting point, whereas boundary optimization subsequently performs fine-grained refinement.

The experiments in Tab. 12 substantiate this claim. We initialize the quantization boundaries using different methods, keep all other settings the same, and train each method for a sufficient number of epochs. The results show that initializing the quantization boundaries with sampling-based estimation offers a clear advantage.

This two-stage paradigm—good initialization followed by fine-grained refinement—is already established in the literature, e.g., in 2DQuant [17].

H.2.5 Generalization to Other Backbones and Tasks

To evaluate whether QuantDemoire generalizes beyond ESDNet, we additionally test FHDe^{2}Net on UHDM under W4A4 quantization. We also evaluate SwinIR on Urban100 for image super-resolution under W4A4 and W8A8 settings. As shown in Tab. 13, QuantDemoire consistently outperforms MinMax and Percentile quantization across both another demoiréing backbone and a Transformer-based super-resolution model.

H.3 Banding Artifacts Visualization

Traditional image quality assessment (IQA) metrics do not accurately reflect the impact of banding artifacts, as illustrated in [29]. Therefore, the advantage of frequency-aware calibration in IQA metrics is not significant. To visually demonstrate the effectiveness of frequency-aware calibration, we present a comparison of the visual results from the model trained with original loss and frequency-aware calibration.

It is evident that under the extremely low bit-width setting of 4-bit, the baseline model (using the original loss) still suffers from severe banding artifacts. In contrast, frequency-aware calibration significantly alleviates this issue.

Method	PSNR	SSIM	LPIPS
Ours (sampling)	20.2803	0.6745	0.3369
percentile 95%	19.4556	0.6223	0.3601
percentile 99.9%	19.5781	0.6391	0.3532
MinMax	18.9265	0.5921	0.4013

Table 12: Comparison of different initialization approaches.

Task / Backbone	Dataset	Method/Bit	PSNR ↑	SSIM ↑
Demoiréing / FHDe^{2}Net	UHDM	MinMax/4	16.42	0.6535
Demoiréing / FHDe^{2}Net	UHDM	Percentile/4	17.86	0.7012
Demoiréing / FHDe^{2}Net	UHDM	QuantDemoire/4	**20.23**	**0.7194**
SR / SwinIR	Urban100	MinMax/4	18.54	0.3871
SR / SwinIR	Urban100	Percentile/4	21.01	0.5179
SR / SwinIR	Urban100	QuantDemoire/4	**24.15**	**0.7324**
SR / SwinIR	Urban100	MinMax/8	21.79	0.7292
SR / SwinIR	Urban100	Percentile/8	21.16	0.7205
SR / SwinIR	Urban100	QuantDemoire/8	**25.96**	**0.8271**
SR / SwinIR	Urban100	FP32/32	26.15	0.8306

Table 13: Generalization results on another demoiréing backbone and image super-resolution.

H.4 Implementation Details of the FP16 branch

Our Mixed-Precision Weight Quantizer retains only 1% of convolutional weights, creating extreme unstructured sparsity that is inherently unfriendly to parallel execution. We address this challenge with a specialized GPU kernel. We employ CSR-style per-channel indexing for sparse weight storage. The computation strategy tiles output rows across CUDA blocks and stripes threads along the column dimension, with each kernel serially traversing the sparse weights and accumulating contributions to the output tensor.

This design choice is justified by the extreme sparsity regime: for a typical 32×16×3×3 kernel, each output channel retains only 1-2 weight elements on average, so serial traversal is more efficient than parallel scheduling. This approach successfully exploits unstructured sparsity while incurring only 6% GPU memory. On an RTX 4090 GPU, our fused CUDA implementation achieves a 2.3× end-to-end speedup over the FP16 model.

H.5 Comprehensive Ablation Study Details

All ablation experiments use static quantization.

Table 1 (a) Quantizer bound calculation: All methods are evaluated without smoothing and without any training-based optimization. Weights are quantized using MinMax. For activation quantization: MinMax uses the absolute minimum and maximum; Percentile applies 99.9th-percentile clipping; Sample uses the sampling-based estimator proposed.

Table 1 (b) Smooth transformation: In this setting, weights are quantized by MinMax, activations are quantized using the sampling-based estimator, and no training-based optimization is applied. Original applies no smoothing; Smooth (Raw) computes the smoothing factor from the absolute maximum activation; Smooth (Sample) computes the smoothing factor using the sampling-based estimator.

Table 2 (a) Outlier handling method: Baseline and Smooth (Sample) in Table 1 (b) share identical settings. Discard Outliers truncates weight outliers; Store Random randomly stores 1% of weights in FP16; Store Outlier stores the largest 0.5% and smallest 0.5% of weights.

Table 2 (b) Outliers Proportion: 0.50% is identical to Store Outlier in Table 2 (a); 0.25% stores the largest 0.25% and smallest 0.25% of weights (all other settings identical to the 0.50% case); 1% stores the largest 1% and smallest 1% of weight elements (others identical to the 0.50%).

Table 3 (a) Calibration strategy: No Calibration uses the same settings as Store Outlier in Table 2 (a). Original Image applies L_1 & L_p calibration on the original images (on top of No Calibration). High-Freq performs calibration using high-frequency image components. Mid&Low-Freq performs calibration using mid- and low-frequency components.

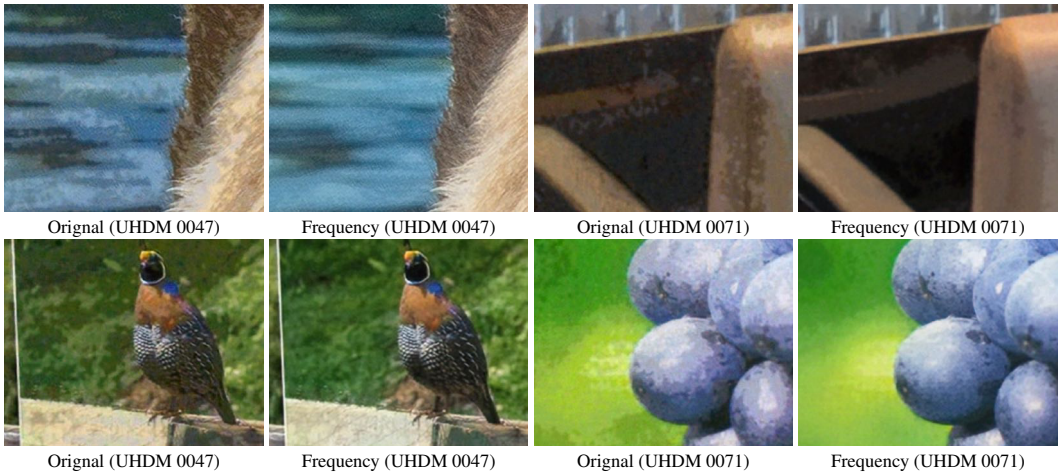

Figure 6: Visualization of the effectiveness of frequency-aware calibration. "Orignal" are the results of model trained with original loss function, "Frequency" are the results of model trained with frequency-aware calibration, which significantly weakened banding artifacts.

Method	PSNR	SSIM	LPIPS	Training epoch
Ours (only learn the clipping bound)	21.0808	0.7626	0.3068	4
Omniquant (learn clipping bound and smoothing factor)	21.0643	0.7613	0.3106	40

Table 14: Comparison of our method vs. Omniquant. Experiments are conducted in 4bit, UHDM.

Table 3 (b) Frequency extraction level: For level = 3, the settings are identical to Mid&Low-Freq in Table 3 (a). For level = 1 and level = 5, we use 1 and 5 convolution kernels, respectively, to extract frequency components; all other settings are the same as for level = 3.

H.6 Comparison with Related Methods

H.6.1 Comparison with OmniQuant

Both the outlier-aware quantizer and OmniQuant [25] follow the same high-level smoothing-and-clipping scheme. However, OmniQuant [25] is designed to quantize large language models (LLMs), whereas the outlier-aware quantizer targets an image demoiréing model composed mainly of convolutional modules. Consequently, the two approaches differ substantially in their specific design choices.

(1) Different clipping targets. OmniQuant [25] employs learnable weight clipping in LWC. By contrast, in convolutional modules the weights are confined to small kernels and are relatively few per layer compared with the volume of activations. Truncating weight outliers can therefore cause disproportionate accuracy loss. Therefore, in the outlier-aware quantizer we clip only the activation quantization bounds rather than clipping the weights.

(2) Choice of learnable parameters. OmniQuant [25] treats both the smoothing factor and the weight clipping bounds as learnable. In our outlier-aware quantization, only the activation clipping bound is learnable; the smoothing factor is initialized using a sampling-based estimator and then kept fixed in the training process. Because the rounding operation in quantization can destabilize training, we observe that our design outperforms the alternative that makes both the smoothing factor and the weight clipping bounds learnable, as the results shown in Tab. 14.

Furthermore, by initializing the smoothing factor to a good value via sampling and reducing the number of learnable parameters, our method requires only 4 training epochs, whereas the configuration with learnable smoothing factor and clipping bounds typically requires 40 epochs. Our approach substantially reduces the training time.



GT Input Original (banding) Mid&Low-Freq

Figure 7: Visualization of results under different calibration strategies. Original: training in the spatial domain suffers from banding artifacts under low-bit quantization. Mid&Low-Freq: our frequency-aware calibration strategy leverages mid- and low-frequency.

H.6.2 Comparison with Mixed-Precision Methods

Many prior methods adopt mixed-precision quantization. For example, LLM.int8 [5] processes a very small number of outlier feature dimensions in FP16, and SqueezeLLM [11] stores a small subset of weights in a sparse format. However, few approaches are designed for convolutions. Because outlier weights in convolution kernels occur sporadically, existing coarse-grained mixed-precision schemes are not well suited, and directly converting convolutions into sparse matrix multiplications introduces additional overhead. To address this, we implement a dedicated kernel for the FP16 branch that parallelizes processing of FP16 weights at the element level.

H.7 Visualization of Different Calibration Strategies

Results under different calibration strategies are shown in Fig. 7. Training directly in the spatial domain (Original) causes the quantized demoiréing model to exhibit severe banding artifacts, which degrades output quality. Our Frequency-Aware Calibration Strategy successfully mitigates banding by optimizing specific frequency components, achieving results close to the ground truth.

I Explanations for Checklist

I.1 Limitations

Despite achieving strong results, QuantDemoire still has several limitations. First, our method requires a representative calibration set; without calibration data, the sampling-based range estimation and frequency-aware calibration cannot be reliably performed. Second, the sampling strategy is most effective when activations and weights contain sparse extreme outliers around a dominant distribution. For layers with highly irregular or strongly multi-modal distributions, the sampling rates may require additional tuning. Third, the current mixed-precision branch relies on a custom CUDA kernel evaluated on an RTX 4090 GPU, and deployment on mobile NPUs or other accelerators may require hardware-specific engineering.

I.2 Broader Impacts

QuantDemoire reduces the memory, computation, and energy cost of image demoiréing models, facilitating their deployment on edge devices such as cameras, drones, and portable imaging systems. This can improve the accessibility of high-quality image restoration under limited computational resources. At the same time, more efficient visual restoration may also be used in privacy-sensitive applications such as surveillance or large-scale data collection, and quantization artifacts under distribution shifts may affect downstream visual analysis. We therefore encourage careful validation on target devices and datasets before deployment.

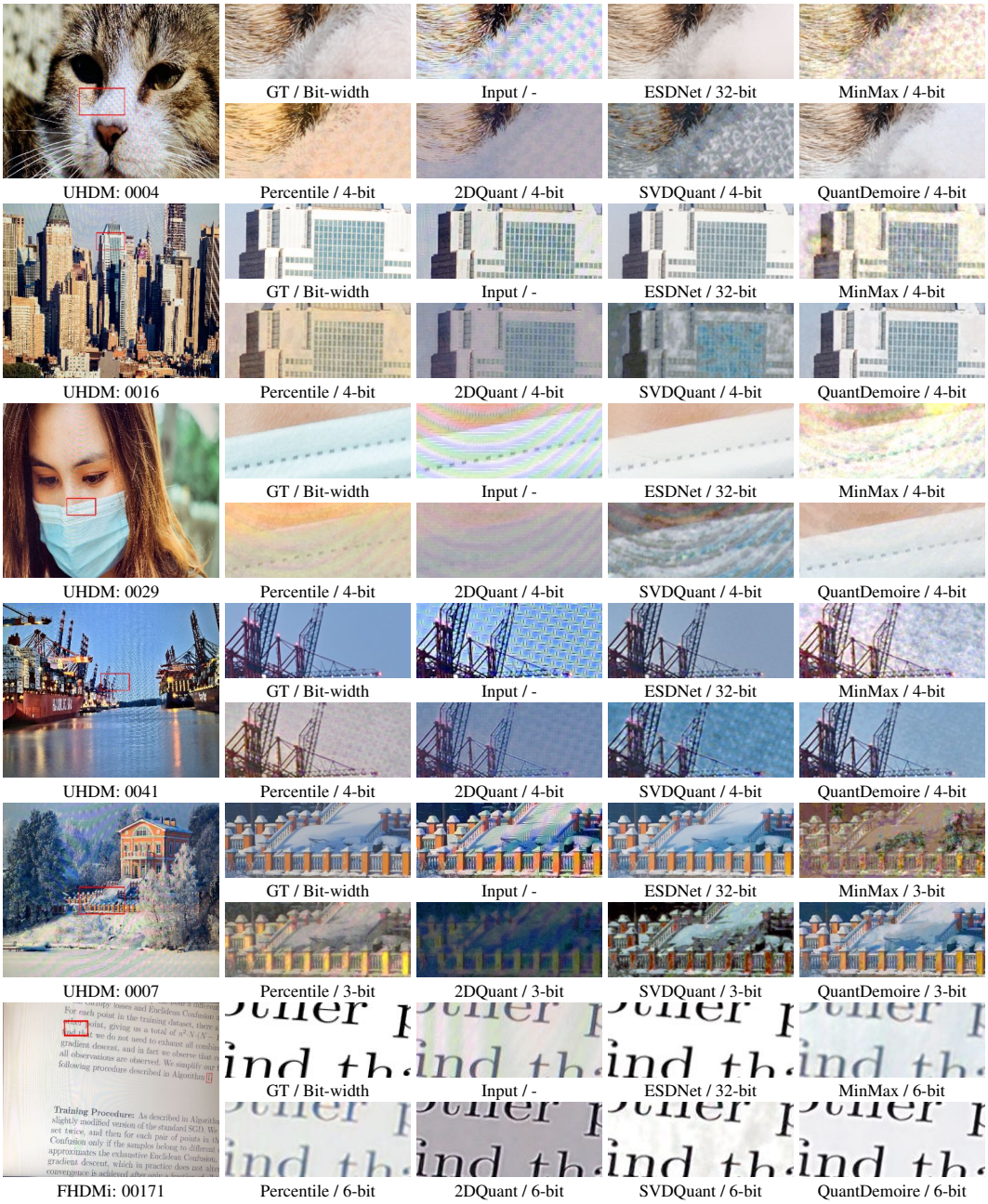

Figure 8: Visual comparison on the UHDM [36] dataset.

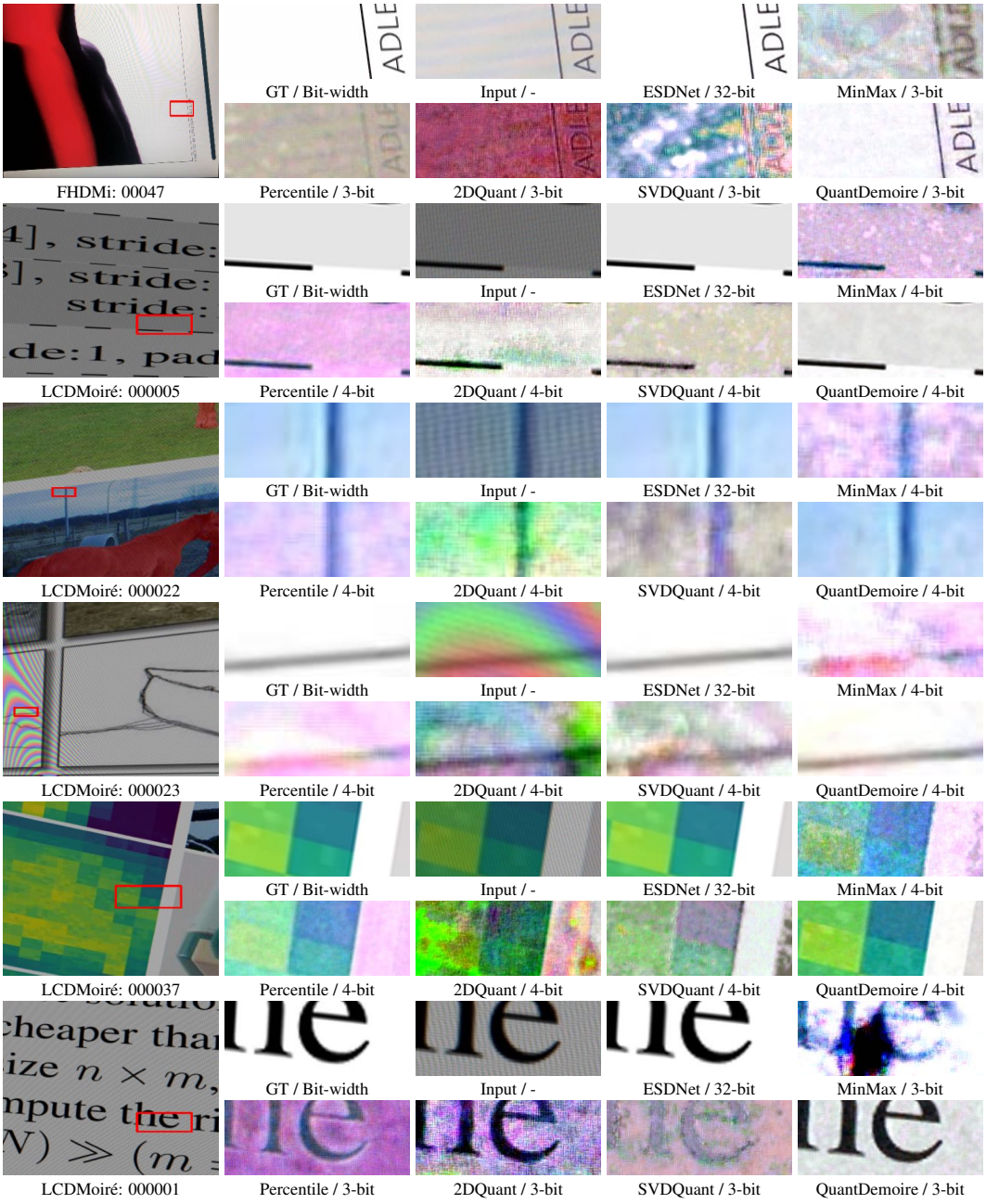

Figure 9: Visual comparison on the LCDMoiré [37] dataset.

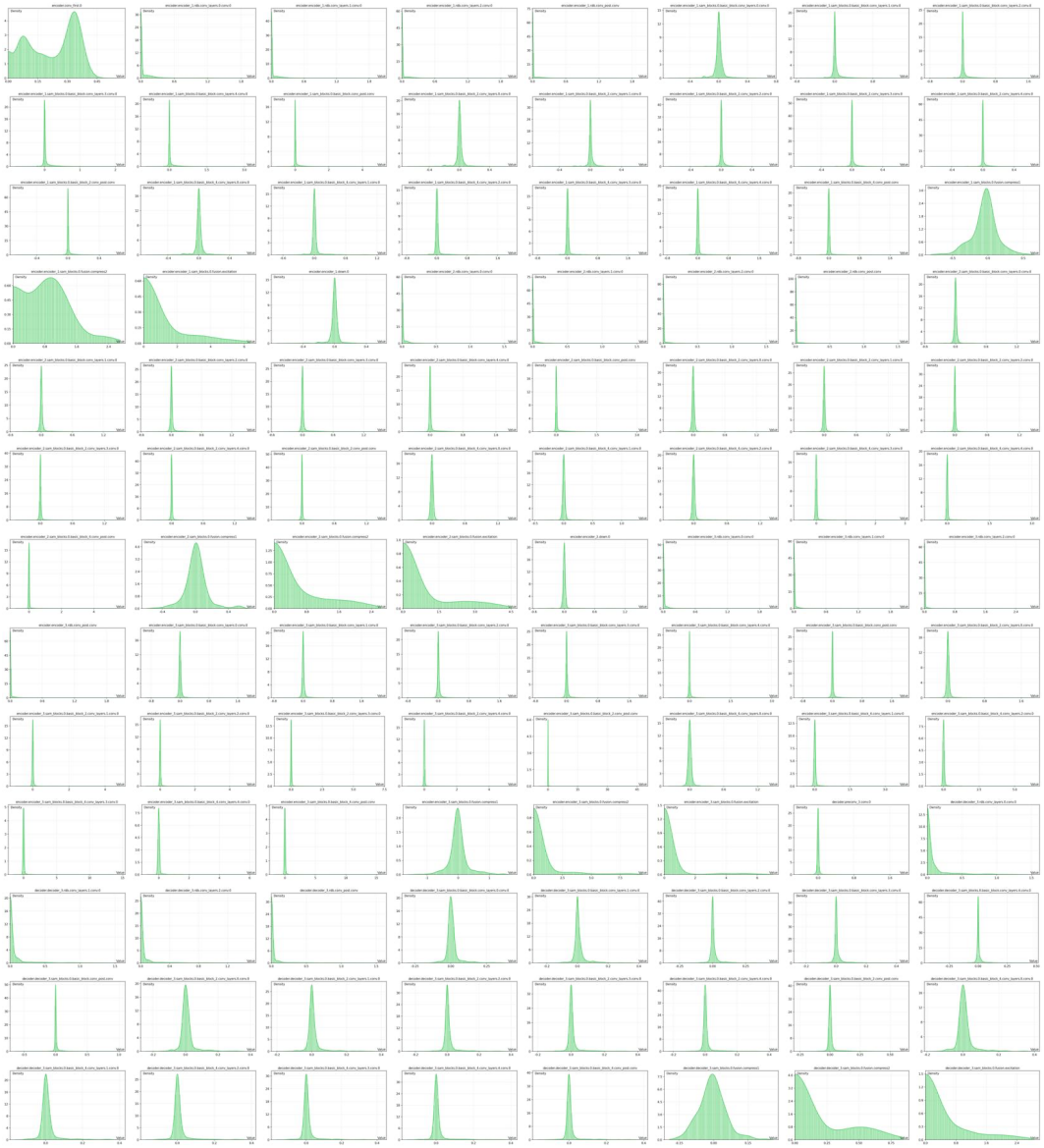

Figure 10: More distribution of activation (Part 1).

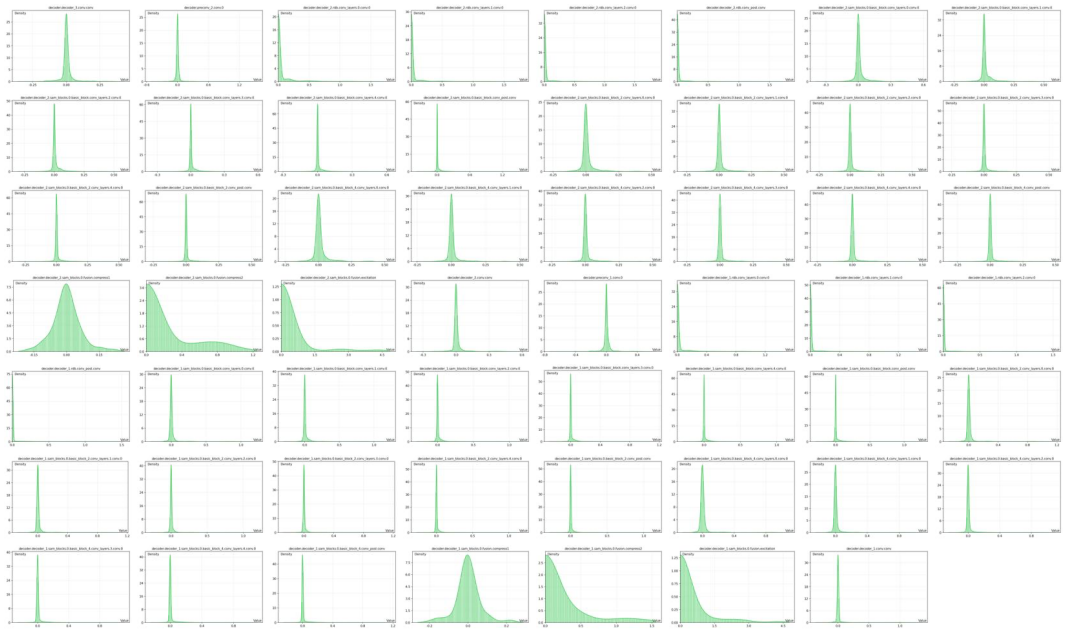

Figure 11: More distribution of activation (Part 2).

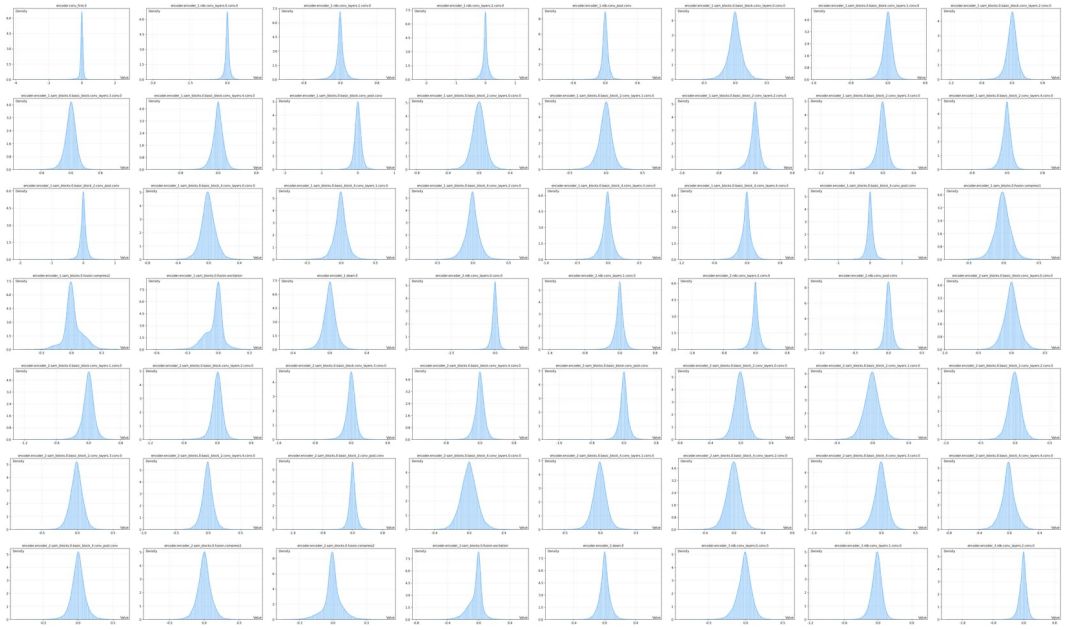

Figure 12: More distribution of weight (Part 1).

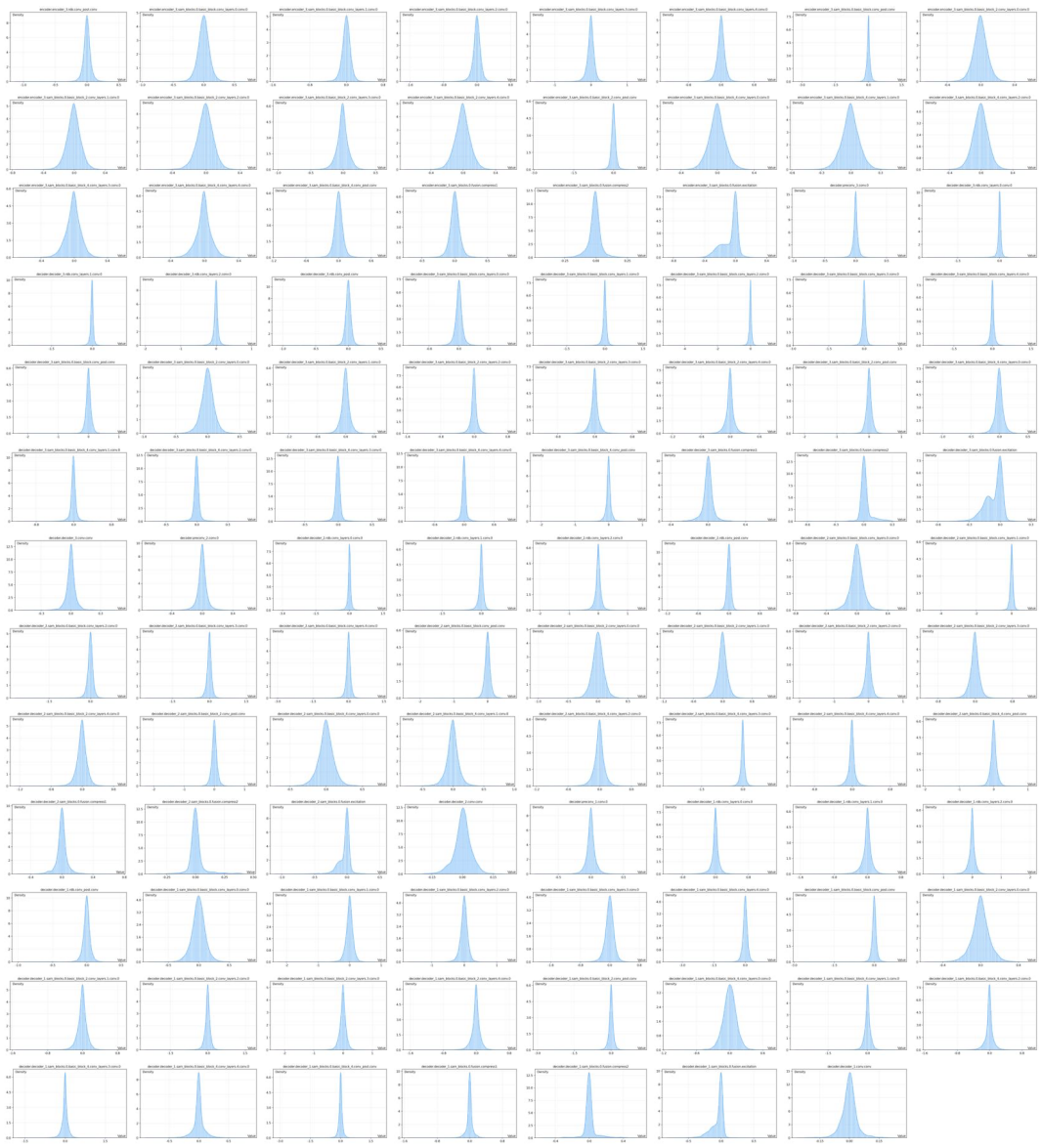

Figure 13: More distribution of weight (Part 2).

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: Please refer to our abstract and introduction.

   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 the limitations in the appendix.

   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 include 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 have provided implementation details in the experiments section. We will also release all the code and models.

   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: [No]

Justification: Firstly, we provide very detailed instructions (*e.g.*, method descriptions and implementation details) to reproduce our dataset and results. Secondly, we promise to release the code, dataset, and all models.

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: We have provided implementation details.

   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: [Yes]

   Justification: Please refer to the experiment part.

   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: Please refer to experiment part.

   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 conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   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: Please refer to the appendix.

    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: This work poses no such risks.

    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 have credited most previous works in the paper. The license and terms are respected properly.

    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: We will release code and models. In the paper, we have provided implementation details and other content to reproduce our results.

    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 nor research with 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 crowdsourcing nor research with 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 were not used as part of the core methodology or experimental pipeline.

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.