2DQuant: Low-bit Post-Training Quantization for Image Super-Resolution

Anonymous Author(s)
Affiliation
Address
email

Abstract

Low-bit quantization has become widespread for compressing image super-resolution (SR) models for edge deployment, which allows advanced SR models to enjoy compact low-bit parameters and efficient integer/bitwise constructions for storage compression and inference acceleration, respectively. However, it is notorious that low-bit quantization degrades the accuracy of SR models compared to their full-precision (FP) counterparts. Despite several efforts to alleviate the degradation, the transformer-based SR model still suffers severe degradation due to its distinctive activation distribution. In this work, we present a dual-stage low-bit post-training quantization (PTQ) method for image super-resolution, namely **2DQuant**, which achieves efficient and accurate SR under low-bit quantization. The proposed method first investigates the weight and activation and finds that the distribution is characterized by coexisting symmetry and asymmetry, long tails. Specifically, we propose Distribution-Oriented Bound Initialization (DOBI), using different searching strategies to search a coarse bound for quantizers. To obtain refined quantizer parameters, we further propose Distillation Quantization Calibration (DQC), which employs a distillation approach to make the quantized model learn from its FP counterpart. Through extensive experiments on different bits and scaling factors, the performance of DOBI can reach the state-of-the-art (SOTA) while after stage two, our method surpasses existing PTQ in both metrics and visual effects. 2DQuant gains an increase in PSNR as high as 4.52dB on Set5 ($\times$2) compared with SOTA when quantized to 2-bit and enjoys a 3.60$\times$ compression ratio and 5.08$\times$ speedup ratio. The code and models will be released.

1 Introduction

As one of the most classical low-level computer vision tasks, image super-resolution (SR) has been widely studied with the significant development of deep neural networks. With the ability to reconstruct high-resolution (HR) image from the corresponding low-resolution (LR) image, SR has been widely used in many real-world scenarios, including medical imaging [13, 21, 19], surveillance [44, 37], remote sensing [1], and mobile phone photography. With massive parameters, DNN-based SR models always require expensive storage and computation in the actual application. Some works have been proposed to reduce the demand for computational power of SE models, like lightweight architecture design and compression. One kind of approach investigates lightweight and efficient models as the backbone for image SR. This progression has moved from the earliest convolutional neural network (CNNs) [10, 11, 25, 47] to Transformers [46, 29, 42, 40, 4, 3] and their combinations. The parameter number significantly decreased while maintaining or even enhancing performance. The other kind of approach is compression, which focuses on reducing the parameter (*e.g.*, pruning and distillation) or bit-width (quantization) of existing SR models.

Model quantization [7, 9, 20, 28] is a technology that compresses the floating-point parameters of a neural network into lower bit-width. The discretized parameters are homogenized into restricted

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

candidate values and cause heterogenization between the FP and quantized models, leading to severe performance degradation. Considering the process, quantization approaches can be divided into quantization-aware training (QAT) and post-training quantization (PTQ). QAT simultaneously optimizes the model parameters and the quantizer parameters [6, 16, 26, 48], allowing them to adapt mutually, thereby more effectively alleviating the degradation caused by quantization. However, QAT often suffers from a heavy training cost and a long training time, and the burden is even much heavier than the training process of the FP counterparts, which necessitates a large amount of compatibility and makes it still far from practical in training-resource-limited scenarios.

Fortunately, post-training quantization emerges as a promising way to quantize models at a low training cost. PTQ fixes the model parameters and only determines the quantizer parameters through search or optimization. Previous researches [39, 26] on PTQ for SR has primarily focused on CNN-based models such as EDSR [30] and SRResNet [24]. However, these quantization methods are not practical for deployment for two reasons. **Firstly**, these CNN-based models themself require huge space and calculation resources. Their poor starting point makes them inferior to advanced models in terms of parameters and computational cost, even after quantization. As shown in Table 1, the light version of SwinIR needs only 16.2% parameters and 15.9% FLOPs compared with quantized EDSR. But its PSNR metric is close to that of the FP EDSR. While the previous PTQ algorithm, DBDC+Pac, suffers from unacceptable degradation in both visual and metrics. **Secondly**, most of these methods can not adapt well to Transformer-based models because of the unadaptable changes in weight and activation distributions. As shown in Figure 1, when applied on SwinIR, the existing methods still suffer from distorted artifacts compared with FP or HR.

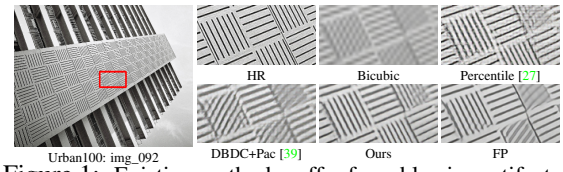

Figure 1: Existing methods suffer from blurring artifacts.

Therefore, we conducted a post-training quantization analysis on super-resolution with a classical Transformer-based model SwinIR [29]. The weight and activation distribution is characterized by coexisting symmetry and asymmetry, long tails. Firstly, if the previous symmetric quantization method is applied for asymmetric distribution, at least half of the candidates are completely ineffective. Besides, the long tail effect causes the vast majority of floating-point numbers to be compressed into one or two candidates, leading to worse parameter homogenization. Furthermore, with such a small number of parameters, SwinIR's information has been highly compressed, and quantizing the model often results in significant performance degradation. Nevertheless, the excellent performance and extremely low computational requirements of Transformer-based models are precisely what is needed for deployment in real-world scenarios.

Table 1: Complexity and performance $(\times 4)$.

Model	EDSR [30]	EDSR (4bit) [39]	SwinIR-light [29]	DBDC+Pac (4bit) [39]	Ours (4bit)
Params (MB)	172.36	21.55	3.42	1.17	1.17
Ops (G)	823.34	103.05	16.74	4.19	4.19
PNSR on Urban100	26.64	25.56	26.47	24.94	25.71

In this paper, we propose **2DQuant**, a two-stage PTQ algorithm for image super-resolution tasks. To enhance the representational capacity in asymmetry scenarios, we employ a quantization method with two bounds. The bounds decide the candidate for numbers out of range and the interval of candidates in range. **First**, we propose **distribution-oriented Bound Initialization** (DOBI), a fast MSE-based searching method. It is designed to minimize the value heterogenization between quantized and FP models. Two different MSE [5] search strategies are applied for different distributions to avoid nonsense traversal. This guarantees minimum value shift while maintaining high speed and efficiency in the search process. **Second**, we propose **Distillation Quantization Calibration** (DQC), a training-based method. It is designed to adjust each bound to its best position finely. This ensures that the outputs and intermediate feature layers of the quantized model and that of the FP model should remain as consistent as possible. Thereby DQC allows the quantizer parameters to be finely optimized toward the task goal. The contributions of this paper can be summarized as follows:

(1) To the best of our knowledge, we are the first to explore PTQ with Transformer-based model in SR thoroughly. We design 2DQuant, a unique and efficient two-stage PTQ method (see Figure 2) for image super-resolution, which utilizes DOBI and DQC to optimize the bound from coarse to fine.

(2) In the first stage of post-quantization, we use DOBI to search for quantizer parameters, employing customized search strategies for different distributions to balance speed and accuracy. In the second stage, we design DQC, a more fine-grained optimization-based training strategy, for the quantized model, ensuring it aligns with the FP model on the calibration set.

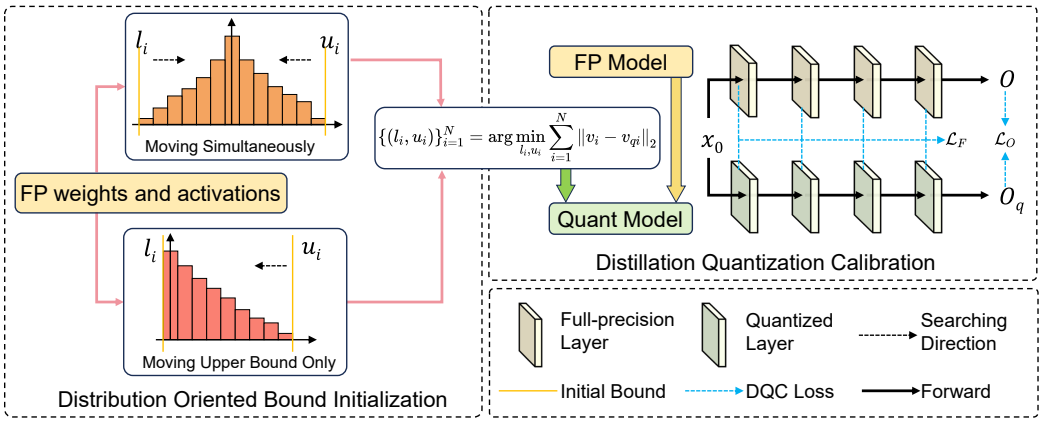

Figure 2: The overall pipeline of the proposed 2DQuant method. The whole pipeline contains two stages, optimizing the clipping bound from coarse to fine. In stage 1, we design DOBI to efficiently obtain the coarse bound. In stage 2, DQC is performed to finetune clipping bounds and guarantee the quantized model learns the FP (FP) model's feature and output information.

(3) Our 2DQuant can compress Transformer-based model to 4,3,2 bits with the compression ratio being 3.07×, 3.31×, and 3.60× and speedup ratio being 3.99×, 4.47×, and 5.08×. No additional module is added so 2DQuant enjoys the theoretical upper limit of compression and speedup.

(4) Through extensive experiments, our 2DQuant surpasses existing SOTA on all benchmarks. We gain an increase in PSNR by as high as 4.52dB in Set5 (×2) when compressed to 2 bits, and our method has a more significant increase when compressed to lower bits.

2 Related Work

Image super-resolution. Deep CNN networks have shown excellent performance in the field of image super-resolution. The earliest SR-CNN [10, 11] method adopted a CNN architecture. It surpassed previous methods in the image super-resolution domain. In 2017, EDSR [30] won the NTIRE2017 [38] championship, becoming a representative work of CNNs in the SR by its excellent performance. Thereafter, with the continuous development of Vision Transformers (ViT) [12], models based on the ViT architecture have surpassed many CNN networks. These Transformer-based models achieve significant performance improvements and they have fewer parameters and lower computational costs. Many works have modified the ViT architecture, achieving continuous improvements. A notable example is SwinIR [29]. With a simple structure, it outperforms many CNN-based models. However, previous explorations of post-quantization in the super-resolution domain have been limited to CNN-based models. They focus on models like EDSR [30] or SRResNet [24]. It is a far cry from advanced models no matter in parameters, FLOPs, or performance. Currently, there is still a research gap in post-quantization for Transformer architectures.

Model quantization. In the field of quantization, quantization methods are mainly divided into PTQ and QAT. QAT is widely accepted due to its minimal performance degradation. PAMS [26] utilizes a trainable truncated parameter to dynamically determine the upper limit of the quantization range. DAQ [17] proposed a channel-wise distribution-aware quantization scheme. CADyQ [16] is proposed as a technique designed for SR networks and optimizes the bit allocation for local regions and layers in the input image. However, QAT usually requires training for as long as or even longer than the original model, which becomes a barrier for real scenarios deployment. Instead of training the model from scratch, existing PTQ methods use the pre-trained models. PTQ algorithms only find the just right clipping bound for quantizers, saving time and costs. DBDC+Pac [39] is the first to optimize the post-training quantization for image super-resolution task. It outperforms other existing PTQ algorithms. Whereas, they only focus on EDSR [30] and SRResNet [24]. Their 4-bit quantized version is inferior to advanced models in terms of parameters and computational cost, let alone performance. It reveals a promising result for PTQ applying on SR, but using a more advanced model could bridge the gap between high-performance models and limited calculation resource scenarios.

3 Methodology

To simulate the precision loss caused by quantization, we use fake-quantize [22],*i.e.*quantization-dequantization, for activations and weights. and the process can be written as

$$v_c = \text{Clip}(v, l, u), \quad v_r = \text{Round}(\frac{2^N - 1}{u - l}(v_c - l)), \quad v_q = \frac{u - l}{2^N - 1} v_r + l, \tag{1}$$

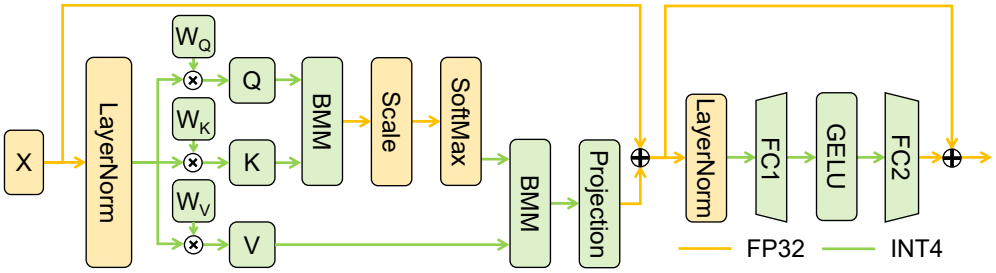

Figure 3: Quantization scheme for SwinIR Transformer blocks. Fake quantization and INT arithmetic are performed in all compute-intensive operators including all linear layers and batch matmul. Lower bits such as 3 or even 2 are also permitted. Dropout of attention and projection is ignored

where v denotes the value being fake quantized, which can be weight or activation. l and u are the lower and upper bounds for clipping, respectively. $\text{Clip}(v, l, u) = \max(\min(v, u), l)$, and Round rounds the input value to the nearest integer. v_c denotes the value after clipping, and v_r denotes the integer representation of v, and v_q denotes the value after fake quantization. The Clip and Round operations contribute to reducing the parameters and FLOPs but also introduce quantization errors.

Figure 3 shows the basic structure of the Transformer block. We have quantized all the modules with a significant computational load within them, effectively reducing the model's FLOPs. Table 2 shows the FLOPs needed for each module. The Linear layers and matrix multiplication account for approximately 86% of the computation load, which are all transformed into integer arithmetic. When performing gradient backpropagation, we follow the Straight-Through Estimator [8] (STE) style:

Table 2: FLOPs distribution.

Module	FLOPs (G)	Ratio (%)
Linear & BMM	14.34	85.66
Conv	2.33	13.90
Other	0.07	0.44
Total	16.74	100.00

$$\frac{\partial v_q}{\partial u} = \frac{\partial v_c}{\partial u} + \frac{1}{2^n - 1} v_r - \frac{v_c - l}{u - l}, \quad \frac{\partial v_q}{\partial l} = \frac{\partial v_c}{\partial l} - \frac{1}{2^n - 1} v_r + \frac{v_c - l}{u - l}, \tag{2}$$

where $\frac{\partial v_c}{\partial u} = H(u - v)$ and $\frac{v_c}{\partial l} = H(l - v)$, $H(\cdot)$ denotes Heaviside step function [45]. This formula approximates the direction of gradient backpropagation, allowing training-based optimization to proceed. The derivation of the formula can be found in the supplementary material.

Figure 2 shows the whole pipeline of 2DQuant, which is a **two**-stage coarse-to-fine post-training quantization method. The first stage is **D**OBI, using **two** strategies to minimize the value shift while the second stage is **D**QC, optimizing **two** bound of each quantizer towards the task goal.

3.1 Analysis of data distribution

To achieve better quantization results, we need to analyze the distribution of the model's weights and activations in detail. We notice that the data distribution shows a significantly different pattern from previous explorations, invalidating many of the previous methods. The weights and activations distribution of SwinIR is shown in Figure 4. More can be found in supplemental material. Specifically, the weights and activations of SwinIR exhibit noticeable long-tail, coexisting symmetry and asymmetry.

Weight. The weights of all linear layers are symmetrically distributed around zero, showing clear symmetry, and are generally similar to a normal distribution. This is attributed to the weight decay applied to weights, which provides quantization-friendly distributions. From the value shift perspective, both symmetric and asymmetric quantization are tolerable. Whereas, from the vantage point of task objectives, asymmetric quantization possesses the potential to offer a markedly enhanced information density, thus elevating the overall precision of the computational processes involved.

Activations. As for activations, they exhibit obvious periodicity in different Transformer Blocks. For V or the input of FC1, the obtained activation values are symmetrically distributed around 0. However, for the attention map or the input of FC2 in each Transformer Block, due to the Softmax calculation or the GELU [14] activation function, the minimum value is almost fixed, and the overall distribution is similar to an exponential distribution. Therefore, the data in SwinIR's weights and activations exhibit two distinctly different distribution characteristics. Setting asymmetric quantization and different search strategies for both can make the search rapid and accurate.

3.2 distribution-oriented bound initialization

Because the data distribution exhibits a significant long-tail effect, we must first clip the range to avoid low effective bits. Common clipping methods include density-based, ratio-based, and MSE-based approaches. The first two require manually specifying the clipping ratio, which significantly affects the clipping outcome and necessitates numerous experiments to determine the optimal ratio. Thus we proposed the Distribution-Oriented Bound Initialization (DOBI) to search the bound for weight and

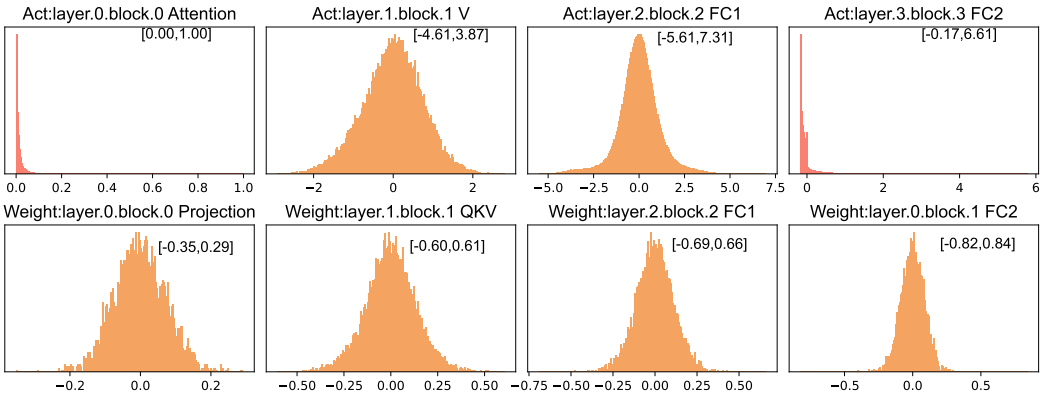

Figure 4: The selected representative distribution of activations (Row 1) and weights (Row 2). The range of data is marked in the figure. All weights obey symmetric distribution. The attention map and the input of FC2 are asymmetric due to softmax function and GELU function.

activation, avoiding manually adjusting hyperparameters. The global optimizing goal is as follows

$$\{(l_i, u_i)\}_{i=1}^{N} = \arg\min_{l_i, u_i} \sum_{i=1}^{N} \|v_i - v_{qi}\|_2 , \tag{3}$$

The collection of all quantizers' bounds $\{(l_i, u_i)\}_{i=1}^{N}$ is the linchpin of quantized model performance as it indicates the candidate value for weights and activations. We note that the data distribution falls into two categories: one resembling a bell-shaped distribution and the other resembling an exponential distribution. For the bell-shaped distribution, we use a symmetric boundary-narrowing search method. Whereas, for the exponential distribution, we fix the lower bound to the minimum value of the data and only traverse the right bound. The specific search method is shown in Algorithm 1. The time complexity of Algorithm 1 is $\mathcal{O}(MK)$, where M is the number of elements in data v and K is the number of search points. The condition *v is symmetrical* is obtained by observing the visualization of v and the activations are from the statistics on a small calibration set.

3.3 Distillation quantization calibration

To further fine-tune the clipping range, we propose distillation quantization calibration (DQC) to transfer the knowledge from the FP model to the quantized model. It leverages the knowledge distillation [15] where the FP model acts as the teacher while the quantized model is the student. Specifically, for the same input image, the student model needs to continuously minimize the discrepancy with the teacher model on the final super-resolution output. The loss for the final output can be written as

$$\mathcal{L}_O = \frac{1}{C_O H_O W_O} \|O - O_q\|_1 , \tag{4}$$

where O and O_q are the final outputs of the teacher and student models, C_O, H_O, and W_O represent the number of output channels, height, and width, respectively. we adopt the L1 loss for the final output, as it tends to converge more easily compared to the L2 loss [30]. As the quantized model shares the same structure with the FP model and is quantized from the FP model, the student model also need to learn to extract the same feature of the teacher model, which can be written as

$$\mathcal{L}_F = \sum_{i}^{N} \frac{1}{C_i H_i W_i} \left\| \frac{F_i}{\|F_i\|_2} - \frac{F_{qi}}{\|F_{qi}\|_2} \right\|_2 , \tag{5}$$

where F_i and F_{qi} are the intermediate features of the teacher and student models respectively and i is the index of the layer. In the field of super-resolution, there is a clear correspondence between the feature maps and the final reconstructed images, making training on feature maps crucial. since the quantized network

Algorithm 1: DOBI pipeline

Data: data to be quantized v, the number of search point K, bit b
Result: Clip bound l, u
$l \leftarrow \min(v)$,$u \leftarrow \max(v)$;
$min\_mse \leftarrow +\infty$;
if v *is symmetrical* **then**
  $\Delta l \leftarrow (\max(v) - \min(v))/2K$;
else
  $\Delta l \leftarrow 0$;
end
$\Delta u \leftarrow (\max(v) - \min(v))/2K$;
while $i \leq K$ **do**
  $l_i \leftarrow l + i \times \Delta l$, $u_i \leftarrow u + i \times \Delta u$;
  get v_q based on Eq. (1);
  $mse \leftarrow \|v - v_q\|_2$;
  if $mse \leq min\_mse$ **then**
    $min\_mse \leftarrow mse$;
    $l\_best \leftarrow l_i$, $u\_best \leftarrow u_i$;
  end
end

and the full-precision network have identical structures, we do not need to add extra adaptation layers for feature distillation. The final loss function can be written as

$$\mathcal{L} = \mathcal{L}_O + \lambda \mathcal{L}_F, \tag{6}$$

where λ is the co-efficient of $\mathcal{L}_F$. In the second stage, based on training optimization methods, the gap between the quantized model and the full-precision model will gradually decrease. The performance of the quantized model will progressively improve and eventually converge to the optimal range.

4 Experiments

4.1 Experimental Settings

Data and Evaluation. We use DF2K [38, 31] as the training data, which combines DIV2K [38] and Flickr2K [31], as utilized by most SR models. During training, since we employ a distillation training method, we do not need to use the high-resolution parts of the DF2K images. For validation, we use the Set5 [2] as the validation set. After selecting the best model, we tested it on five commonly used benchmarks in the SR field: Set5 [2], Set14 [43], B100 [34], Urban100 [18], and Manga109 [35]. On the benchmarks, we input low-resolution images into the quantized model to obtain reconstructed images, which we then compared with the high-resolution images to calculate the metrics. We do not use self-ensemble in the test stage as it increases the computational load eightfold, but the improvement in metrics is minimal The evaluation metrics we used are the most common metrics PSNR and SSIM [41], which are calculated on the Y channel (*i.e.*, luminance) of the YCbCr space.

Implementation Details. We use SwinIR-light [29] as the backbone and provide its structure in the supplementary materials. We conduct comprehensive experiments with scale factors of 2, 3, and 4 and with 2, 3, and 4 bits, where Our hyperparameter settings remain consistent. During DOBI, we use a search step number of K=100, and the statistics of activations are obtained from 32 images in DF2K being randomly cropped to retain only $3\times64\times64$. During DQC, we use the Adam [23] optimizer with a learning rate of 1×10^{-2}, betas set to (0.9, 0.999), and a weight decay of 0. We employ CosineAnnealing [33] as the learning rate scheduler to stabilize the training process. Data augmentation is also performed. We randomly utilize rotation of 90°, 180°, and 270° and horizontal flips to augment the input image. The total iteration for training is 3,000 with batch size of 32. Our code is written with Python and PyTorch [36] and runs on an NVIDIA A800-80G GPU.

4.2 Comparison with State-of-the-Art Methods

The methods we compared include MinMax [22], Percentile [27], and the current SOTA post-quantization method in the super-resolution field, DBDC+Pac [39]. For a fair comparison, we evaluated the performance of DBDC+Pac [39] on EDSR [30], as the authors performed detailed parameter adjustments and model training on EDSR, and we directly used the results reported by the authors, recorded in the table as EDSR. It should be noted that the EDSR method uses self-ensemble in the final test, which can improve performance to some extent but comes at the cost of 8 times the computational load. Additionally, we applied DBDC+Pac [39] to SwinIR, using the same hyperparameters as those set by the authors for EDSR, recorded in the table as DBDC+Pac [39]. The following are the quantitative and qualitative results of the comparison.

Quantitative Results. Table 3 shows the extensive results of comparing different quantization methods with bit depths of 2, 3, and 4, as well as different scaling factors of $\times2$, $\times3$, and $\times4$.

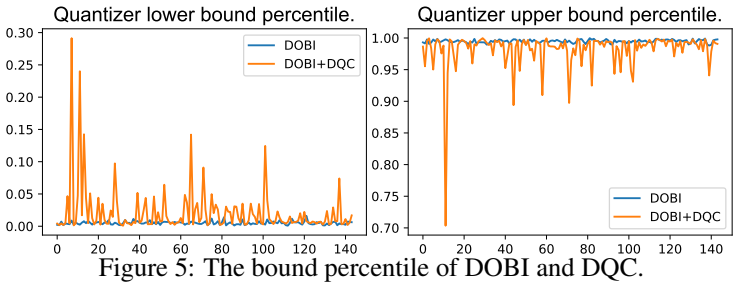

Figure 5: The bound percentile of DOBI and DQC.

DBDC+Pac [39] performs poorly mainly because **1.** The DBDC process requires manually specifying the clipping ratio, which significantly affects performance. **2.** DBDC does not prune weights, and the learning rate in the Pac process is too low, causing slow convergence of weight quantizer parameters. However, both adverse factors are eliminated in our 2DQuant algorithm. When using only DOBI algorithm, our performance has already reached a level comparable to that of DBDC+Pac algorithms. Upon applying DQC, our performance experienced a remarkable and discernible enhancement, elevating it to new heights. In the case of $\times2$, 4-bit on Set5 and Urban100, DOBI has an improvement of 1.11dB and 0.39 dB compared to EDSR, while

Table 3: Quantitative comparison with SOTA methods.

Method	Bit	Set5 (×2)		Set14 (×2)		B100 (×2)		Urban100 (×2)		Manga109 (×2)	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Baseline	32	38.15	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.11	0.9781
Bicubic	32	32.25	0.9118	29.25	0.8406	28.68	0.8104	25.96	0.8088	29.17	0.9128
MinMax [22]	4	34.39	0.9202	30.55	0.8512	29.72	0.8409	28.40	0.8520	33.70	0.9411
Percentile [27]	4	37.37	0.9568	32.96	0.9113	31.61	0.8917	31.17	0.9180	37.19	0.9714
EDSR [30, 39]	4	36.33	0.9420	32.75	0.9040	31.48	0.8840	30.90	0.9130	N/A	N/A
DBDC+Pac [39]	4	37.18	0.9550	32.86	0.9106	31.56	0.8908	30.66	0.9110	36.76	0.9692
DOBI (Ours)	4	37.44	0.9568	33.15	0.9132	31.75	0.8937	31.29	0.9193	37.93	0.9743
2DQuant (Ours)	4	37.87	0.9594	33.41	0.9161	32.02	0.8971	31.84	0.9251	38.31	0.9761
MinMax [22]	3	28.19	0.6961	26.40	0.6478	25.83	0.6225	25.19	0.6773	28.97	0.7740
Percentile [27]	3	34.37	0.9170	31.04	0.8646	29.82	0.8339	28.25	0.8417	33.43	0.9214
DBDC+Pac [39]	3	35.07	0.9350	31.52	0.8873	30.47	0.8665	28.44	0.8709	34.01	0.9487
DOBI (Ours)	3	36.37	0.9496	32.33	0.9041	31.12	0.8836	29.65	0.8967	36.18	0.9661
2DQuant (Ours)	3	37.32	0.9567	32.85	0.9106	31.60	0.8911	30.45	0.9086	37.24	0.9722
MinMax [22]	2	33.88	0.9185	30.81	0.8748	29.99	0.8535	27.48	0.8501	31.86	0.9306
Percentile [27]	2	30.82	0.8016	28.80	0.7616	27.95	0.7232	26.30	0.7378	30.37	0.8351
DBDC+Pac [39]	2	34.55	0.9386	31.12	0.8912	30.27	0.8706	27.63	0.8649	32.15	0.9467
DOBI (Ours)	2	35.25	0.9361	31.72	0.8917	30.62	0.8699	28.52	0.8727	34.65	0.9529
2DQuant (Ours)	2	36.00	0.9497	31.98	0.9012	30.91	0.8810	28.62	0.8819	34.40	0.9602

Method	Bit	Set5 (×3)		Set14 (×3)		B100 (×3)		Urban100 (×3)		Manga109 (×3)	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Baseline	32	34.63	0.9290	30.54	0.8464	29.20	0.8082	28.66	0.8624	33.99	0.9478
Bicubic	32	29.54	0.8516	27.04	0.7551	26.78	0.7187	24.00	0.7144	26.16	0.8384
MinMax [22]	4	31.66	0.8784	28.17	0.7641	27.19	0.7257	25.60	0.7485	29.98	0.8854
Percentile [27]	4	33.34	0.9137	29.61	0.8275	28.49	0.7899	27.06	0.8242	32.10	0.9303
DBDC+Pac [39]	4	33.42	0.9143	29.69	0.8261	28.51	0.7869	27.05	0.8217	31.89	0.9274
DOBI (Ours)	4	33.78	0.9200	29.87	0.8338	28.72	0.7970	27.53	0.8391	32.57	0.9367
2DQuant (Ours)	4	34.06	0.9231	30.12	0.8374	28.89	0.7988	27.69	0.8405	32.88	0.9389
MinMax [22]	3	26.01	0.6260	23.41	0.4944	22.46	0.4182	21.70	0.4730	24.68	0.6224
Percentile [27]	3	30.91	0.8426	28.02	0.7545	27.23	0.7183	25.32	0.7349	29.43	0.8537
DBDC+Pac [39]	3	30.91	0.8445	28.02	0.7538	26.99	0.6937	25.10	0.7122	28.84	0.8403
DOBI (Ours)	3	32.85	0.9075	29.33	0.8200	28.27	0.7820	26.36	0.8036	31.14	0.9178
2DQuant (Ours)	3	33.24	0.9135	29.56	0.8255	28.50	0.7873	26.65	0.8116	31.46	0.9235
MinMax [22]	2	26.05	0.5827	24.74	0.5302	24.42	0.4973	22.87	0.5155	24.66	0.5652
Percentile [27]	2	25.30	0.5677	23.60	0.4890	23.77	0.4751	22.33	0.4965	24.65	0.5882
DBDC+Pac [39]	2	29.96	0.8254	27.53	0.7507	27.05	0.7136	24.57	0.7117	27.23	0.8213
DOBI (Ours)	2	30.54	0.8321	27.74	0.7312	26.69	0.6643	24.80	0.6797	28.18	0.7993
2DQuant (Ours)	2	31.62	0.8887	28.54	0.8038	27.85	0.7679	25.30	0.7685	28.46	0.8814

Method	Bit	Set5 (×4)		Set14 (×4)		B100 (×4)		Urban100 (×4)		Manga109 (×4)	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Baseline	32	32.45	0.8976	28.77	0.7858	27.69	0.7406	26.48	0.7980	30.92	0.9150
Bicubic	32	27.56	0.7896	25.51	0.6820	25.54	0.6466	22.68	0.6352	24.19	0.7670
MinMax [22]	4	28.63	0.7891	25.73	0.6657	25.10	0.6061	23.07	0.6216	26.97	0.8104
Percentile [27]	4	30.64	0.8679	27.61	0.7563	26.96	0.7151	24.96	0.7479	28.78	0.8803
EDSR [30, 39]	4	31.20	0.8670	27.98	0.7600	27.09	0.7140	25.56	0.7640	N/A	N/A
DBDC+Pac [39]	4	30.74	0.8609	27.66	0.7526	26.97	0.7104	24.94	0.7369	28.52	0.8697
DOBI (Ours)	4	31.10	0.8770	28.03	0.7672	27.18	0.7237	25.43	0.7631	29.31	0.8916
2DQuant (Ours)	4	31.77	0.8867	28.30	0.7733	27.37	0.7278	25.71	0.7712	29.71	0.8972
MinMax [22]	3	19.41	0.3385	18.35	0.2549	18.79	0.2434	17.88	0.2825	19.13	0.3097
Percentile [27]	3	27.55	0.7270	25.15	0.6043	24.45	0.5333	22.80	0.5833	26.15	0.7569
DBDC+Pac [39]	3	27.91	0.7250	25.86	0.6451	25.65	0.6239	23.45	0.6249	26.03	0.7321
DOBI (Ours)	3	29.59	0.8237	26.87	0.7156	26.24	0.6735	24.17	0.6880	27.62	0.8349
2DQuant (Ours)	3	30.90	0.8704	27.75	0.7571	26.99	0.7126	24.85	0.7355	28.21	0.8683
MinMax [22]	2	23.96	0.4950	22.92	0.4407	22.70	0.3943	21.16	0.4053	22.94	0.5178
Percentile [27]	2	23.03	0.4772	22.12	0.4059	21.83	0.3816	20.45	0.3951	20.88	0.3948
DBDC+Pac [39]	2	25.01	0.5554	23.82	0.4995	23.64	0.4544	21.84	0.4631	23.63	0.5854
DOBI (Ours)	2	28.82	0.7699	26.46	0.6804	25.97	0.6319	23.67	0.6407	26.32	0.7718
2DQuant (Ours)	2	29.53	0.8372	26.86	0.7322	26.46	0.6927	23.84	0.6912	26.07	0.8163

2DQuant has an improvement of 0.69 dB and 1.18 dB compared to the SOTA method. All these results indicate that our two-stage PTQ method can effectively mitigate the degradation caused by quantization and ensure the quality of the reconstructed images.

Figure 5 shows the bound percentile of DOBI searching and DQC. Overall, the bound of DQC is tighter as the values around the zero point enjoy greater importance. Besides, the shallow layers' bounds vary more significantly due to the elevated significance of these layers within the neural

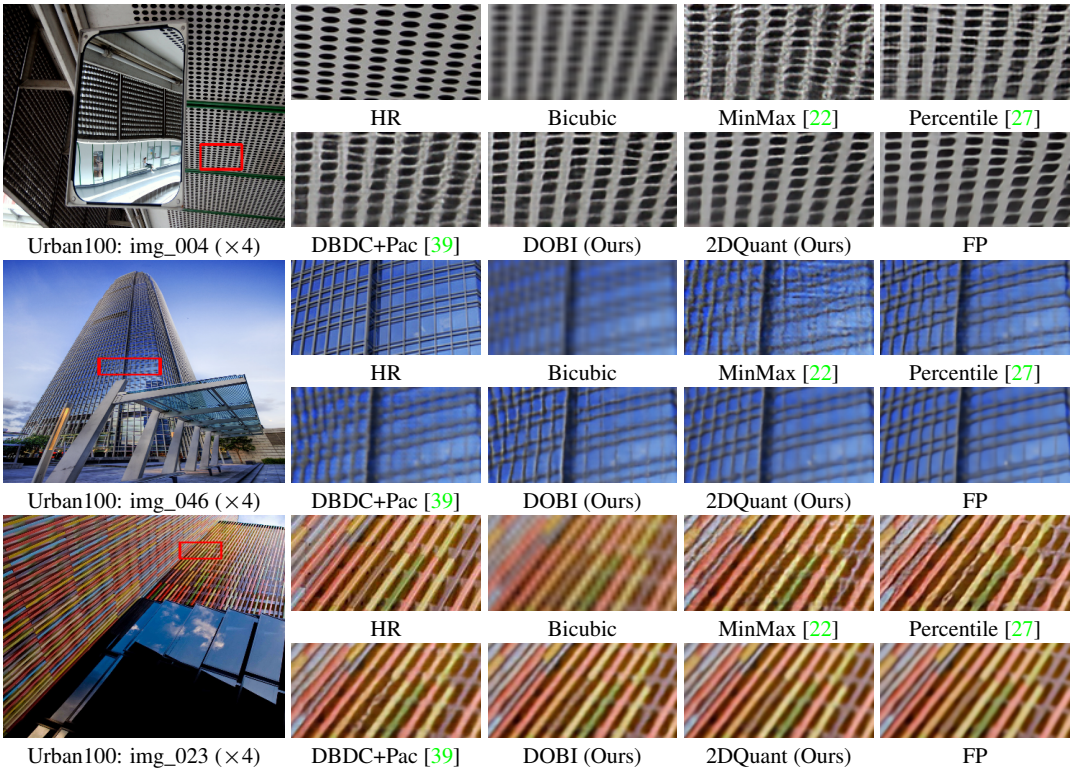

Figure 6: Visual comparison for image SR (×4) in some challenging cases.

Learning rate	PSNR↑	SSIM↑
10^{-1}	37.82	0.9594
10^{-2}	37.87	0.9594
10^{-3}	37.78	0.9592
10^{-4}	37.74	0.9587

(a) Learning rate

Batch size	PSNR↑	SSIM↑
4	37.82	0.9594
8	37.83	0.9594
16	37.84	0.9593
32	37.87	0.9594

(b) Batch size

DOBI	DQC	PSNR↑	SSIM↑
		34.39	0.9202
✓		37.44	0.9568
	✓	37.32	0.9563
✓	✓	37.87	0.9594

(c) DOBI and DQC

Table 4: Ablation studies. The models are trained on DIV2K and Flickr2K, and tested on Set5 (×2)

network. Detailedly, the bound for the second MLP fully connected layer's weight in Layer 0 Block 1 only remains 46% data in its range. It has the second-highest lower bound percentile and the smallest upper bound percentile among the network. Its percentiles are 0.2401 and 0.7035 respectively while its bound values are -0.062 and 0.047 and its distribution is visualized in Figure 4. In conclusion, only through task-oriented optimization of each bound at a fine-grained level can redundant information be maximally excluded and useful information be maximally retained.

Qualitative Results. We show the visual comparison results for ×4 in Figure 12. Since quantized models are derived from full-precision models with information loss, their global performance will rarely exceed that of full-precision models. As seen in the three images for Minmax, after quantization, if no clipping is performed, the long tail effect will lead to a large number of useless bits, resulting in a significant amount of noise and repeated distorted patterns in the reconstructed images. In these challenging cases, our training method allows the model to retain edge information of objects better, preventing blurring and distorted effects. For example, in img_046 and img_023, we have the highest similarity to the full-precision model, while other methods show varying degrees of edge diffusion, significantly affecting image quality. Compared to the DBDC+Pac method, our DOBI and DQC allow for better representation of edge and texture information in the images and effectively avoid distortions and misalignments in the graphics. The visual results demonstrate that our proposed DQC is essential for improving performance in both metric and visual comparisons.

4.3 Ablation Study

Learning Rate and Batchsize. We first study the performance variations of the model under different hyperparameters. From Tables 4a and 4b, it can be seen that our DQC enables the model to

converge within a range of outstanding performance for most learning rates and batch sizes. Due to the non-smooth impact of quantization parameters on the model, the quantized model is more prone to local optima compared to the full-precision model, resulting in a noticeable performance drop when the learning rate is too low. Additionally, as shown in Table 4b, the larger the batch size, the better the model's performance, and the smoother the convergence process. However, even with a smaller batch size, we can still achieve a performance of 37.82dB on Set5, indicating that our two-stage method has good robustness to different hyperparameters.

DOBI and DQC. Moreover, we also study the impact of different stages on performance, with the results shown in Table 4c, from which we can draw the following conclusions: **Firstly**, the goal of DOBI is to minimize the value shift for weights and activations. Although it is not the task goal, it can still enjoy significant enhancement due to better bit representational ability. **Secondly**, DQC alone cannot achieve the optimization effect of DOBI. This is because the impact of quantizer parameters on model performance is oscillatory, and training alone is prone to converge to local optima. In contrast, search-based methods can naturally avoid local optima. So it's necessary to use results from the search-based method to initialize training-based method in PTQ. **Thirdly**, when DOBI and DQC are combined, namely our 2DQuant, the 4-bit quantized model has only a 0.28dB decrease on Set5 compared to the FP model, which maximally mitigates the accuracy loss caused by quantization.

5 Discussion

Why our results surpass FP outcomes While our method's performance metrics do not yet fully match those of full-precision models, visual results reveal a compelling advantage. As observed in image img_092 of Figure 1 of Urban100, our approach correctly identifies the direction of the stripes in the image. Whereas the full-precision model erroneously selects the wrong direction. This discrepancy arises because the lower-resolution image, affected by aliasing, creates an illusion of slanted stripes, misleading the FP model's reconstruction. This phenomenon demonstrates that our PTQ algorithm allows more accurate restored results in certain localized and challenging tasks without being misled. More examples are in the supplementary materials.

It suggests that full-precision models contain not only redundant knowledge but also incorrect information. The latter is hard to get rid of by training the FP model. Our quantization method can effectively reduce model parameters and computational demands while eliminating erroneous information, achieving multiple benefits simultaneously. This also suggests that the FP model doesn't represent the pinnacle of what a quantized model can achieve.

Limitations. Despite achieving excellent results, this study still has some limitations. During the DOBI process, the data distribution of activations and weights is required to approximate a bell curve or exponential distribution; otherwise, the DOBI method cannot find the most suitable positions. Additionally, increasing the number of search points for a single tensor in MSE does not necessarily guarantee better performance. However, the second-stage training can somewhat alleviate this issue. Moreover, our method requires a calibration set; without which, the first-stage DOBI and the second-stage DQC cannot be carried out at all.

Societal Impacts. Our super-resolution quantization method effectively saves computational resources, facilitating the deployment of super-resolution models at the cutting edge

6 Conclusion

This paper studies the post-training quantization in the field of image super-resolution. We first conducted a detailed analysis of the data distribution of Transformer-based model in SR. These data exhibit a clear long-tail effect and symmetry and asymmetry coexisting effect. We designed 2DQuant, a dual-stage PTQ algorithms. In the first stage DOBI, we designed two different search strategies for the two different distributions. In the second stage DQC, we designed a distillation-based training method that let the quantized model learn from the FP model, minimizing the accuracy loss caused by quantization. Our 2DQuant can compress Transformer-based model to 4,3,2 bits with the compression ratio being 3.07×, 3.31×, and 3.60× and speedup ratio being 3.99×, 4.47×, and 5.08×. No additional module is added so 2DQuant enjoys the theoretical upper limit of compression and speedup. Extensive experiments demonstrate that 2DQuant surpasses all existing PTQ methods in the field of SR and even surpasses the FP model in some challenging cases. In the future, recognizing the significant impact of the model on performance, we will conduct PTQ research on more advanced super-resolution models and attempt to deploy quantized super-resolution algorithms to actual photography tasks, providing a more detailed evaluation of the performance of PTQ algorithms.

References

[1] Wele Gedara Chaminda Bandara and Vishal M. Patel. Hypertransformer: A textural and spectral feature fusion transformer for pansharpening. In *CVPR*, 2022. 1

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, 2012. 6

[3] Zheng Chen, Yulun Zhang, Jinjin Gu, Linghe Kong, Xiaokang Yang, and Fisher Yu. Dual aggregation transformer for image super-resolution. In *CVPR*, 2023. 1

[4] Zheng Chen, Yulun Zhang, Jinjin Gu, Yongbing Zhang, Linghe Kong, and Xin Yuan. Cross aggregation transformer for image restoration. In *NeurIPS*, 2022. 1

[5] Jungwook Choi, Pierce I-Jen Chuang, Zhuo Wang, Swagath Venkataramani, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Bridging the accuracy gap for 2-bit quantized neural networks (qnn). *arXiv*, 2018. 2

[6] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 2

[7] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *ICCVW*, 2019. 1

[8] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016. 4, 18

[9] Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *ACM MM*, 2022. 1

[10] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 1, 3

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2016. 1, 3

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, 2020. 3

[13] Hayit Greenspan. Super-resolution in medical imaging. *The Computer Journal*, 2008. 1

[14] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, 2016. 4

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Workshop*, 2014. 5

[16] Cheeun Hong, Sungyong Baik, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Cadyq: Content-aware dynamic quantization for image super-resolution. In *ECCV*, 2022. 2, 3

[17] Cheeun Hong, Heewon Kim, Sungyong Baik, Junghun Oh, and Kyoung Mu Lee. Daq: Channel-wise distribution-aware quantization for deep image super-resolution networks. In *WACV*, 2022. 3

[18] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 6

[19] Yawen Huang, Ling Shao, and Alejandro F Frangi. Simultaneous super-resolution and cross-modality synthesis of 3d medical images using weakly-supervised joint convolutional sparse coding. In *CVPR*, 2017. 1

[20] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training quantization with small calibration sets. In *ICML*, 2021. 1

[21] Jithin Saji Isaac and Ramesh Kulkarni. Super resolution techniques for medical image processing. In *ICTSD*, 2015. 1

[22] 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. 3, 6, 7, 8, 19

[23] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2, 3

[25] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 1

[26] Huixia Li, Chenqian Yan, Shaohui Lin, Xiawu Zheng, Baochang Zhang, Fan Yang, and Rongrong Ji. Pams: Quantized super-resolution via parameterized max scale. In *ECCV*, 2020. 2, 3

[27] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *CVPR*, 2019. 2, 6, 7, 8, 19

[28] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. In *ICLR*, 2021. 1

[29] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCVW*, 2021. 1, 2, 3, 6

[30] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 2, 3, 5, 6, 7

[31] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 6

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 14

[33] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv*, 2016. 6

[34] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 6

[35] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 2017. 6

[36] 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. *NeurIPS*, 2019. 6

[37] Pejman Rasti, Tõnis Uiboupin, Sergio Escalera, and Gholamreza Anbarjafari. Convolutional neural network super resolution for face recognition in surveillance monitoring. In *AMDO*, 2016. 1

[38] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, 2017. 3, 6

[39] Zhijun Tu, Jie Hu, Hanting Chen, and Yunhe Wang. Toward accurate post-training quantization for image super resolution. In *CVPR*, 2023. 2, 3, 6, 7, 8, 19

[40] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, 2022. 1

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 6

[42] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 1

[43] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Proc. 7th Int. Conf. Curves Surf.*, 2010. 6

[44] Liangpei Zhang, Hongyan Zhang, Huanfeng Shen, and Pingxiang Li. A super-resolution reconstruction algorithm for surveillance images. *Elsevier Signal Processing*, 2010. 1

[45] Weihong Zhang and Ying Zhou. Chapter 2 - level-set functions and parametric functions. In *The Feature-Driven Method for Structural Optimization*. Elsevier. 4

[46] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 1

[47] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 1

[48] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. 2

Appendix / supplemental material

A Detailed structure of SwinIR

SwinIR comprises three core modules: shallow feature extraction, deep feature extraction, and high-quality (HQ) image reconstruction.

Shallow and Deep Feature Extraction. Given a low-quality (LQ) input $I_{LQ} \in \mathbb{R}^{H\times W\times C_{in}}$ (where H, W, and C_{in} represent the image height, width, and input channel number, respectively), a 3×3 convolutional layer $H_{SF}(\cdot)$ is employed to extract shallow features $F_0 \in \mathbb{R}^{H\times W\times C}$ as follows:

$$F_0 = H_{SF}(I_{LQ}), \tag{7}$$

where C denotes the number of feature channels. Subsequently, deep features $F_{DF} \in \mathbb{R}^{H\times W\times C}$ are extracted from F_0 as:

$$F_{DF} = H_{DF}(F_0), \tag{8}$$

where $H_{DF}(\cdot)$ represents the deep feature extraction module, comprising K residual Swin Transformer blocks (RSTB) and a 3×3 convolutional layer. Specifically, intermediate features $F_1, F_2, \ldots, F_K$ and the output deep feature F_{DF} are sequentially extracted as follows:

$$\begin{aligned} F_i &= H_{RSTB_i}(F_{i-1}), \quad i = 1, 2, \ldots, K, \\ F_{DF} &= H_{CONV}(F_K), \end{aligned} \tag{9}$$

where $H_{RSTB_i}(\cdot)$ denotes the i-th RSTB, and H_{CONV} is the concluding convolutional layer. Incorporating a convolutional layer at the end of feature extraction introduces the inductive bias of the convolution operation into the Transformer-based network, laying a robust foundation for subsequent aggregation of shallow and deep features.

Image Reconstruction. In the context of image super-resolution (SR), the high-quality image I_{RHQ} is reconstructed by combining shallow and deep features as follows:

$$I_{RHQ} = H_{REC}(F_0 + F_{DF}), \tag{10}$$

where $H_{REC}(\cdot)$ is the reconstruction module's function. The reconstruction module is implemented using a sub-pixel convolution layer to upsample the feature.Additionally, residual learning is utilized to reconstruct the residual between the LQ and HQ images instead of the HQ image itself, formulated as:

$$I_{RHQ} = H_{SwinIR}(I_{LQ}) + I_{LQ}, \tag{11}$$

where $H_{SwinIR}(\cdot)$ represents the SwinIR function.

A.1 Residual Swin Transformer Block

The residual Swin Transformer block (RSTB) is a residual block incorporating Swin Transformer layers (STL) and convolutional layers. Given the input feature $F_{i,0}$ of the i-th RSTB, intermediate features $F_{i,1}, F_{i,2}, \ldots, F_{i,L}$ are first extracted by L Swin Transformer layers as follows:

$$F_{i,j} = H_{STL_{i,j}}(F_{i,j-1}), \quad j = 1, 2, \ldots, L, \tag{12}$$

where $H_{STL_{i,j}}(\cdot)$ is the j-th Swin Transformer layer in the i-th RSTB. A convolutional layer is added before the residual connection, and the output of RSTB is formulated as:

$$F_{i,out} = H_{CONV_i}(F_{i,L}) + F_{i,0}, \tag{13}$$

where $H_{CONV_i}(\cdot)$ is the convolutional layer in the i-th RSTB.

Swin Transformer Layer. Given an input of size $H \times W \times C$, the Swin Transformer first reshapes the input into a $\frac{HW}{M^2} \times M^2 \times C$ feature by partitioning the input into non-overlapping $M \times M$ local windows, where $\frac{HW}{M^2}$ is the total number of windows. It then computes the standard self-attention for each window (i.e., local attention). For a local window feature $X \in \mathbb{R}^{M^2\times C}$, the *query*, *key*, and *value* matrices Q, K, and V are computed as follows:

$$Q = XP_Q, \quad K = XP_K, \quad V = XP_V, \tag{14}$$

where P_Q, P_K, and P_V are projection matrices shared across different windows. Typically, $Q, K, V \in \mathbb{R}^{M^2\times d}$. The attention matrix is then computed via the self-attention mechanism within a local window as follows:

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \tag{15}$$

where B is the learnable relative positional encoding. In practice, the attention function is performed h times in parallel, and the results are concatenated for multi-head self-attention (MSA).

Next, a multi-layer perceptron (MLP) with two fully-connected layers and GELU non-linearity between them is used for further feature transformations. The LayerNorm (LN) layer is added before both MSA and MLP, with residual connections employed for both modules. The entire process is formulated as:

$$\begin{aligned} X &= \text{MSA}(\text{LN}(X)) + X, \\ X &= \text{MLP}(\text{LN}(X)) + X. \end{aligned} \tag{16}$$

However, when the partition is fixed across different layers, there are no connections between local windows. Thus, regular and shifted window partitioning are used alternately to enable cross-window connections [32], with shifted window partitioning involving shifting the feature by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ pixels before partitioning.

A.2 Our settings

We use the SwinIR light version provided by the original authors. The light version has only 4 RSTBs in the body part while for each RSTB, there are only 6 STLs. For each STL's MSA, the number of heads is 6, the embedding dimension is 60, the window size is 8, and the MLP ratio is 2.

B Detailed distribution of weights and activations

In code implementation, the RSTB is called layers while the STL is called blocks. We visualize all layers' distribution of the pre-trained SwinIR light model's weights in Figure 7. Bias is ignored as it is not quantized. Also, we visualize the distribution of activations from 32 image patches with a size of $3\times64\times64$ in Figure 8, Figure 9, Figure 10, and Figure 11.

We can safely ignore the detailed value of each axis but just care about the shape of distributions.

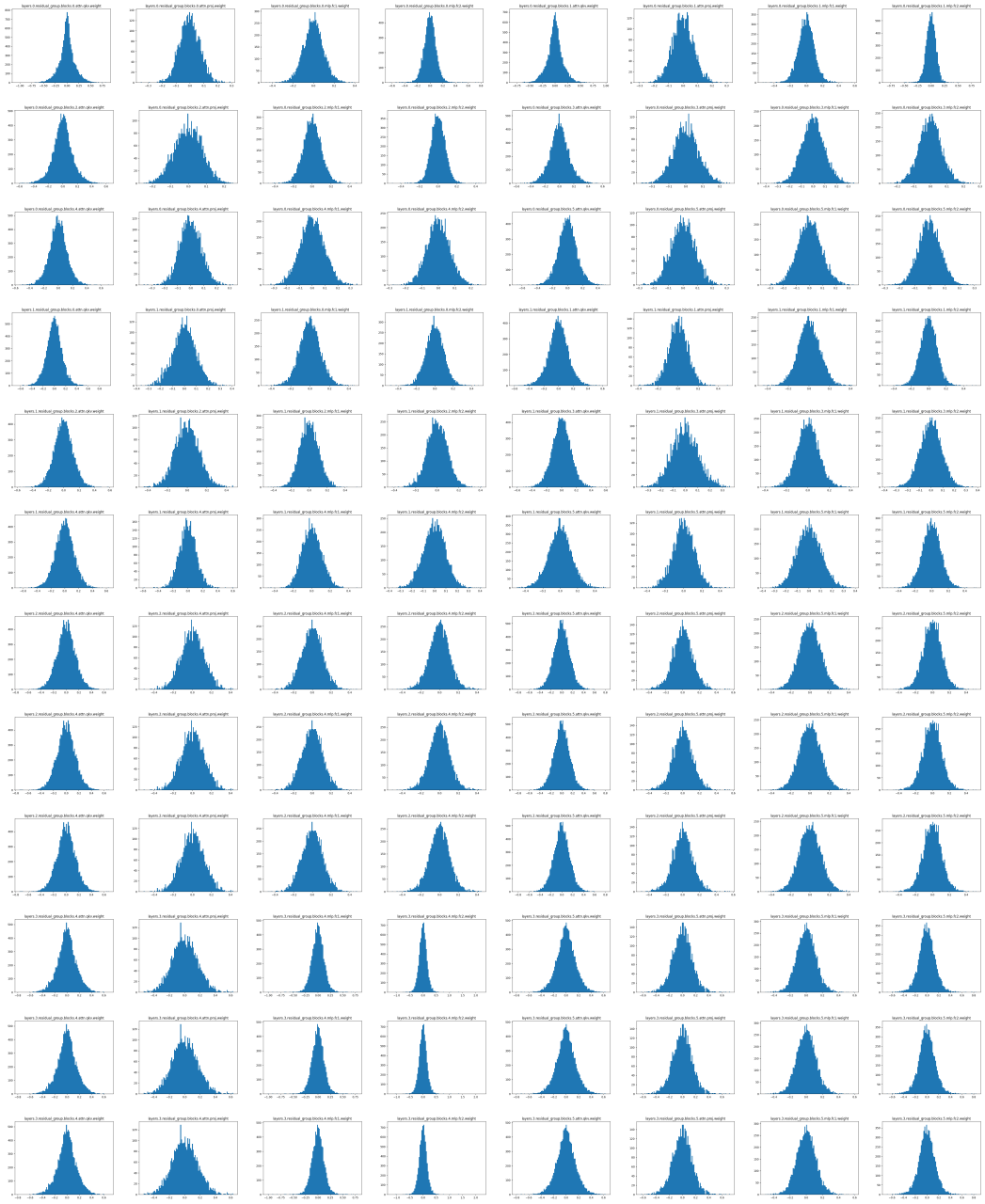

Figure 7: Visualization of SwinIR weights.

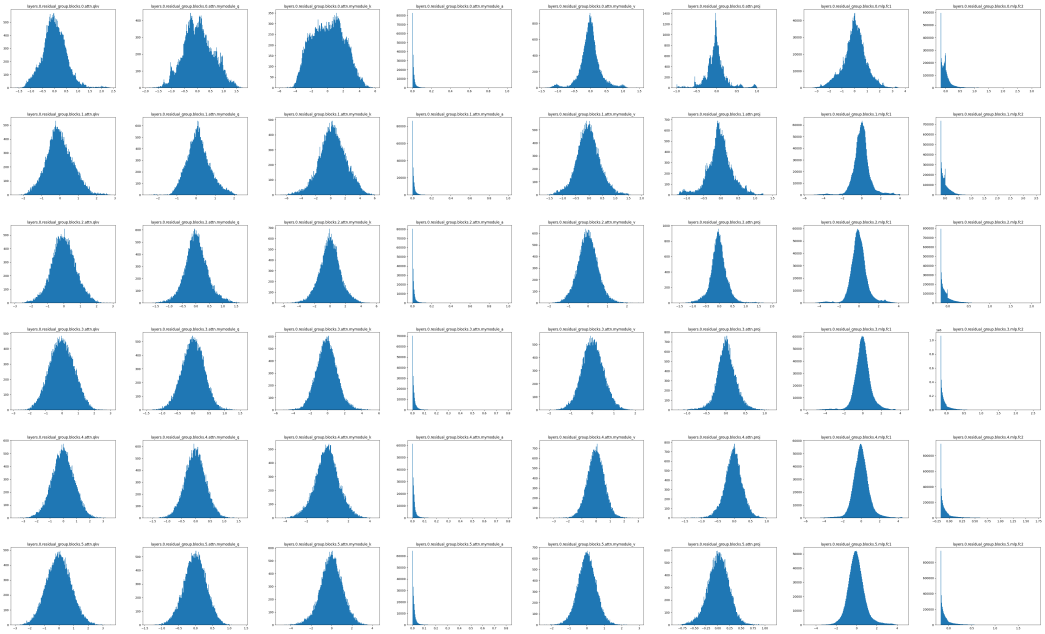

Figure 8: Visualization of SwinIR first layer activation.

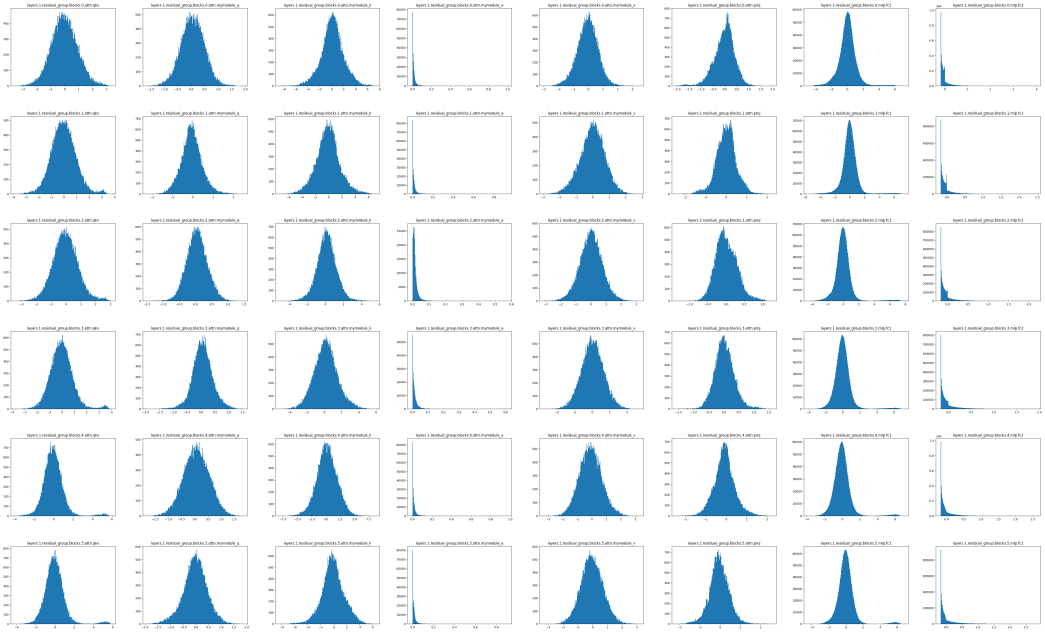

Figure 9: Visualization of SwinIR second layer activation.

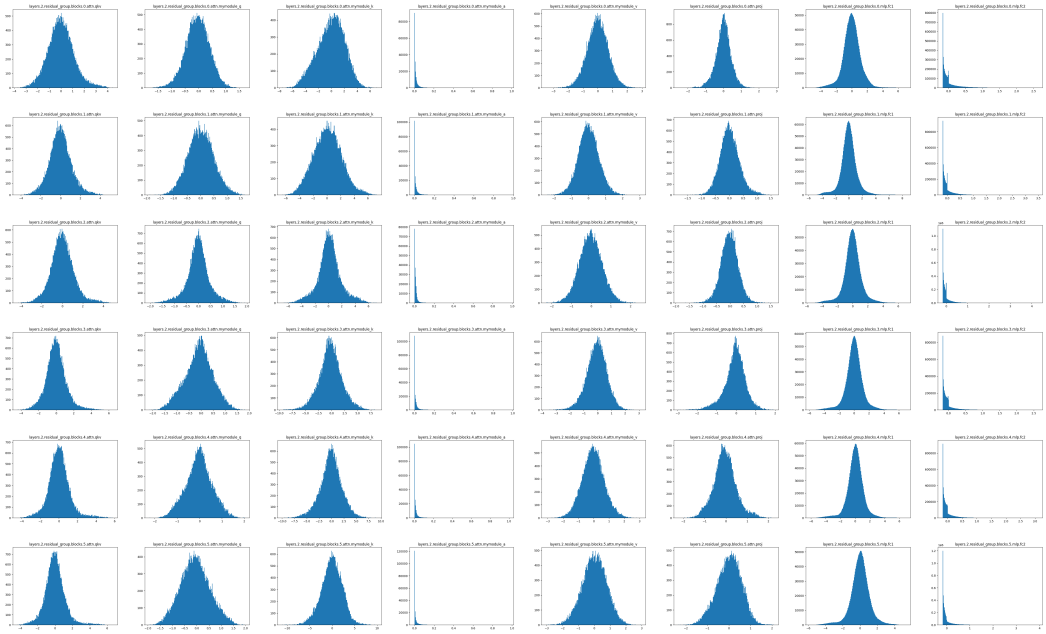

Figure 10: Visualization of SwinIR third layer activation.

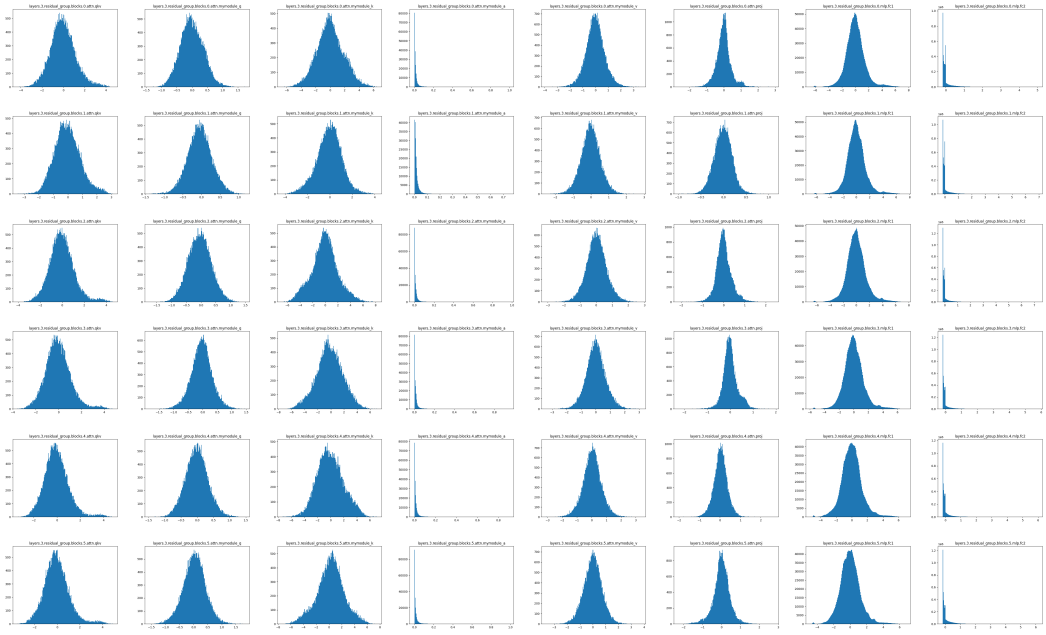

Figure 11: Visualization of SwinIR fourth layer activation.

C The derivation of the backward gradient propagation formula

In this section, we provide the derivation of our backpropagation formula.We follow the STE [8] style to process the round term, which is

$$\frac{\partial \text{Round (x)}}{\partial x} = 1 \tag{17}$$

As for the clip function, we take a similar approach, which is

$$\begin{aligned}
\frac{\partial \text{Clip}(x,l,u)}{\partial x} &= \begin{cases} 1 & \text{if } l \le x \le u \\ 0 & \text{if } x < l \text{ or } x > u \end{cases} \\
\frac{\partial \text{Clip}(x,l,u)}{\partial l} &= \begin{cases} 1 & \text{if } x < l \\ 0 & \text{if } x \ge l \end{cases} \\
\frac{\partial \text{Clip}(x,l,u)}{\partial u} &= \begin{cases} 1 & \text{if } x > u \\ 0 & \text{if } x \le u \end{cases}
\end{aligned} \tag{18}$$

With Eq. (1),Eq. (C), and Eq. (C), we first derive $\frac{\partial v_q}{\partial u}$

$$\begin{aligned}
\frac{\partial v_q}{\partial u} &= \frac{\partial}{\partial u}(\frac{u-l}{2^N-1}v_r + l) \\
&= \frac{1}{2^N-1}v_r + \frac{u-l}{2^N-1}\frac{\partial v_r}{\partial u} \\
&= \frac{1}{2^N-1}v_r + \frac{u-l}{2^N-1}(-\frac{2^N-1}{(u-l)^2}(v_c-l) + \frac{2^N-1}{u-l}\frac{\partial v_c}{\partial u}) \\
&= \frac{\partial v_c}{\partial u} + \frac{1}{2^n-1}v_r - \frac{v_c-l}{u-l}
\end{aligned} \tag{19}$$

$\frac{\partial v_q}{\partial l}$ can be derived roughly the same, which can be written as

$$\begin{aligned}
\frac{\partial v_q}{\partial l} &= \frac{\partial}{\partial l}(\frac{u-l}{2^N-1}v_r + l) \\
&= -\frac{1}{2^N-1}v_r + \frac{u-l}{2^N-1}\frac{\partial v_r}{\partial u} + 1 \\
&= -\frac{1}{2^N-1}v_r + \frac{u-l}{2^N-1}(\frac{2^N-1}{(u-l)^2}(v_c-l) + \frac{2^N-1}{u-l}(\frac{\partial v_c}{\partial u} - 1)) + 1 \\
&= \frac{\partial v_c}{\partial u} - \frac{1}{2^n-1}v_r + \frac{v_c-l}{u-l}
\end{aligned} \tag{20}$$

D More visual examples

We provide more visual illustrations to demonstrate the superiority of our method, as shown in Figure X. In img_016, our method does not distort straight lines. In img_040, our method does not introduce noise to the camera and does not alter the shape at the camera lens. In img_072, we once again outperform the full-precision model by not adding vertical stripes to the curtains. In img_096, we ensure the shape of each window to the greatest extent. These images prove that we can surpass the current SOTA methods in visual effects and avoid misleading results in some tricky cases, generating correct results.

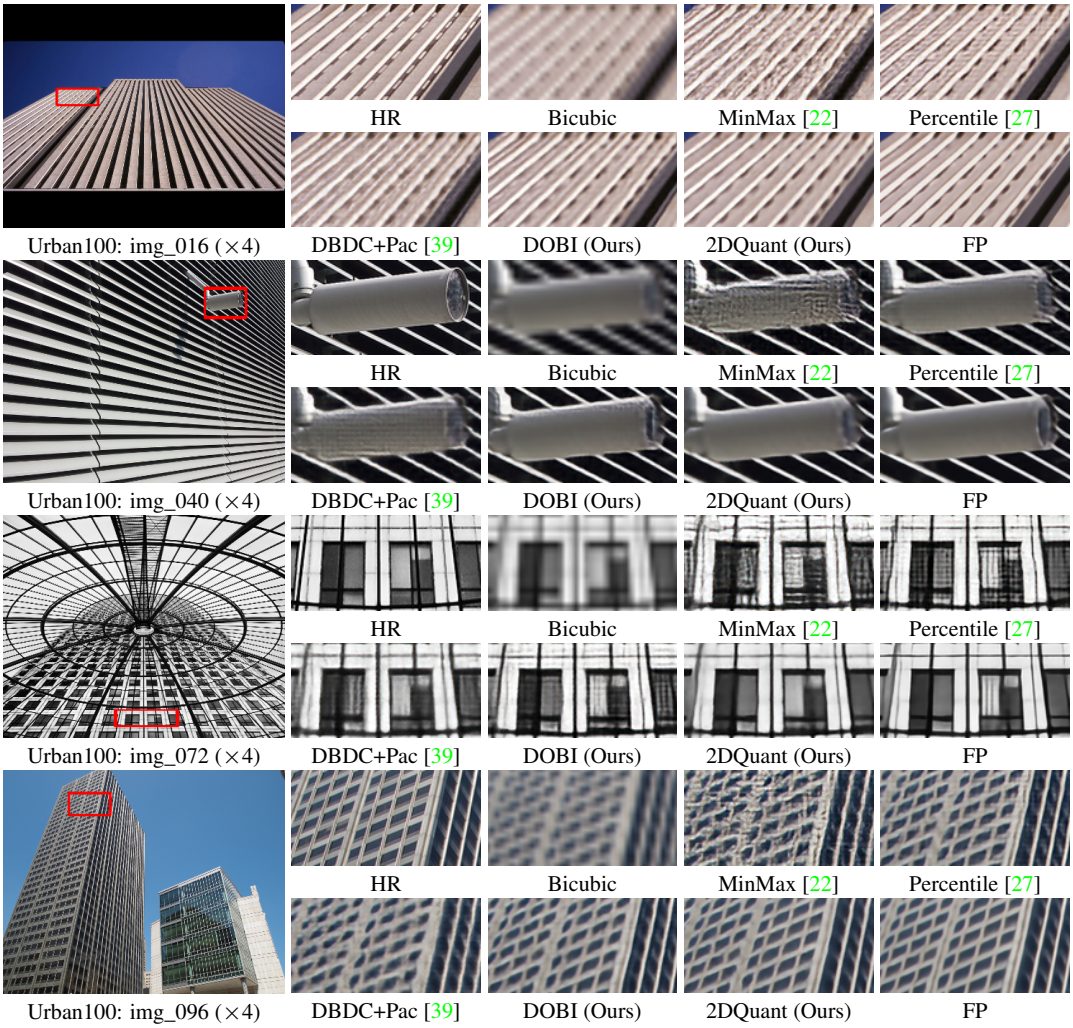

Figure 12: Visual comparison for image SR (×4) in some challenging cases.

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 NA 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 NA 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 main paper.

   Guidelines:

   - The answer NA 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: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA 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 NA 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, all data in the paper can be accessed publicly. Secondly, we provide very detailed instructions (*e.g.*, method descriptions and implementation details) to reproduce our results. Thirdly, we promise to release code and all models.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided implementation details, which cover the above questions.

   Guidelines:

   - The answer NA 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 NA 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 NA 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 NA 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 supplementary file.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This work poses no such risks.

    Guidelines:

    - The answer NA 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 NA 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 contents to reproduce our results.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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.