

---

# QuantSR: Accurate Low-bit Quantization for Efficient Image Super-Resolution

---

Haotong Qin<sup>\*1,2</sup>, Yulun Zhang<sup>\*✉2</sup>, Yifu Ding<sup>1</sup>, Yifan Liu<sup>2</sup>,  
Xianglong Liu<sup>✉1</sup>, Martin Danelljan<sup>2</sup>, Fisher Yu<sup>2</sup>  
<sup>1</sup>Beihang University <sup>2</sup>ETH Zürich

## Abstract

Low-bit quantization in image super-resolution (SR) has attracted copious attention in recent research due to its ability to reduce parameters and operations significantly. However, many quantized SR models suffer from accuracy degradation compared to their full-precision counterparts, especially at ultra-low bit widths (2-4 bits), limiting their practical applications. To address this issue, we propose a novel quantized image SR network, called QuantSR, which achieves accurate and efficient SR processing under low-bit quantization. To overcome the representation homogeneity caused by quantization in the network, we introduce the *Redistribution-driven Learnable Quantizer* (RLQ). This is accomplished through an inference-agnostic efficient redistribution design, which adds additional information in both forward and backward passes to improve the representation ability of quantized networks. Furthermore, to achieve flexible inference and break the upper limit of accuracy, we propose the *Depth-dynamic Quantized Architecture* (DQA). Our DQA allows for the trade-off between efficiency and accuracy during inference through weight sharing. Our comprehensive experiments show that QuantSR outperforms existing state-of-the-art quantized SR networks in terms of accuracy while also providing more competitive computational efficiency. In addition, we demonstrate the scheme's satisfactory architecture generality by providing QuantSR-C and QuantSR-T for both convolution and Transformer versions, respectively. Our code and models are released at <https://github.com/htqin/QuantSR>.

## 1 Introduction

Single image super-resolution (SR) is the task of obtaining a high-resolution (HR) version of a low-resolution (LR) image by retrieving high-frequency details. It is an ill-posed problem since there are multiple HR candidates for the same LR input. To address this issue, researchers have explored the use of deep neural networks, including convolutional neural networks (CNNs) and Transformers [18, 26, 38, 41, 25, 3, 39, 43], to achieve high-quality reconstruction. However, existing SR models rely on expensive computational resources, which significantly limits the real-world SR applications on resource-constrained edge devices. Therefore, there is an urgent requirement to develop model compression techniques for SR models to reduce the computational overhead.

Model quantization [5, 14, 24, 8] has emerged as a powerful compression technique that compresses weights and activations in computing units (such as convolutional and linear layers) into low-bit representations. This results in heavy floating-point operations being converted into efficient integer ones, making quantization highly desirable for edge devices and friendly for memristor-based hardware. Meanwhile, the quantization process primarily focuses on parameter compression within operations. It is considered architecture- and task-agnostic, making it popular among the deep learning

---

\* Equal contribution ✉ Corresponding authors: Yulun Zhang and Xianglong Liu

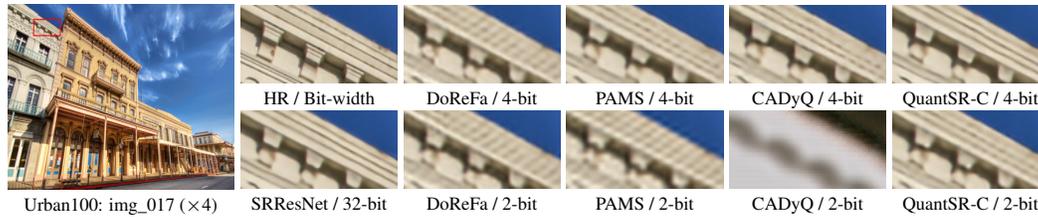


Figure 1: Visual comparison ( $\times 4$ ) with quantized lightweight SR models in terms of 4-bit and 2-bit. We use SRResNet [20] as the full-precision SR backbone and quantize it with low bit width. We compare our QuantSR-C with recent quantization methods (*i.e.*, DoReFa [44], PAMS [22], and CADyQ [10]). Our QuantSR-C performs obviously better than others in both 4-bit and 2-bit cases.

community. Additionally, the quantization function, known as the quantizer, can be customized for each bit width (usually 2-8 bits) to achieve a flexible balance between accuracy and efficiency.

Although quantization is known to enhance the efficiency of image SR models, it can also lead to significant performance degradation, particularly when using ultra-low bit width, *e.g.*, 2-4 bits. It should be noted that another extreme 1-bit setting (also known as binarization) [7, 28] is not considered here. This is mainly because 1-bit quantization suffers from a much larger performance gap and has a different hardware implementation in practice when compared with low-bit quantization settings. Despite attempts to minimize the loss compared to their full-precision counterparts, a considerable performance gap still exists in existing low-bit quantized SR models.

We identify two main reasons for this performance degradation. **Firstly**, the quantizer aims to compress the parameters by discretizing them. It results in the homogenization of the parameter representation in the SR model and causes the loss of gradient information during the backward pass. **Secondly**, the current quantized SR models are specific bit-width mappings of the original full-precision counterparts, and the accuracy of the latter determines its upper performance bound. And once trained, the quantized SR model loses the ability to balance accuracy and efficiency when deployed, limiting its flexibility in practice usage. In this work, we set out to address these problems, to achieve efficient, yet accurate, flexible quantized image SR models.

We propose an accurate **Quantized** model for **Super-Resolution (QuantSR)**. We first propose a method called *Redistribution-driven Learnable Quantizer (RLQ)* that mitigates the representation homogenization caused by discretization. We achieve this by designing a redistributable and learnable quantizer that improves the forward and backward information. Our RLQ results in significant diversification of the representation without incurring additional inference burden. We further propose a *Depth-dynamic Quantized Architecture (DQA)* that advances the performance of quantized SR and provides flexibility in inference resource utilization. We construct weight-sharing deep-dynamic models from an SR architecture base with higher accuracy upper bound, enabling us to achieve superior performance and resource adaptation during inference.

Our comprehensive experiments show that QuantSR outperforms existing quantized SR models across various bit widths (see Fig. 1) by a substantial margin. Notably, our QuantSR with 4 bits achieves performance comparable to existing methods using 8 bits. We also demonstrate the effectiveness of our QuantSR on CNN- and Transformer-based SR networks, highlighting its versatility across different architectures. Our main contributions are summarized as follows:

- We propose QuantSR, a novel accurate quantization scheme for efficient image SR. It provides the potential to deploy the SR application on edge devices and further narrow the performance gap between the quantized model and its full-precision counterpart.
- We propose a Redistribution-driven Learnable Quantizer (RLQ). Specifically, our RLQ diversifies quantized representation and gradient information by redistribution in quantizers. RLQ greatly enhances representations of quantized SR models with little inference burden.
- We propose a Depth-dynamic Quantized Architecture (DQA) to achieve superior performance with the same network depth. Such a dynamic strategy further allows for multiple models with different depths, which is resource adaptation during inference.
- We employ our QuantSR to compress CNN- and Transformer- based SR networks to lower bit-width, resulting in the corresponding quantized baselines, QuantSR-C and QuantSR-T. Our QuantSR achieves superior performance over SOTA quantized SR methods.

## 2 Related Work

### 2.1 Efficient Image Super-Resolution (SR)

Recently, there has been a growing interest in efficient image super-resolution (SR) models due to their resource-friendly characteristics. Typically, researchers aim to develop lightweight networks through various methods, such as architecture design, neural architecture search (NAS), knowledge distillation (KD), and network pruning. For instance, Ahn *et al.* proposed a cascading method using a residual network [1]. While Hui *et al.* designed an information multi-distillation network [15]. Additionally, model compression techniques have also been applied to lightweight SR, such as NAS in FALSr [6], KD employed in training lighter SR student networks [9, 21], and network pruning in ASSL [42]. These techniques have obtained promising results, but they either overlook the fine-grained parameter redundancy or require dramatically additional computations, leaving significant room for compression from a bit-width perspective (*i.e.*, low-bit quantization).

### 2.2 Low-bit Quantization for SR Networks

There exist two primary methods in the quantization field: Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT). PTQ has gained popularity due to its capability to quantize models without necessitating retraining. But, the absence of training and the fixed parameters in the pre-trained model constrain its potential for achieving extreme low-bit quantization [5, 17, 14, 24, 8]. Fortunately, QAT provides us with an opportunity to leverage the entire training process to achieve aggressive low-bit quantization and has shown promising performance [22, 10, 44, 4]. This method enables more comprehensive model optimization and allows for optimal training of the quantized model. QAT is thus considered powerful to achieve extremely low-bit quantization, *e.g.*, 2-4 bits.

The practical utility of existing SR networks on resource-limited devices is hindered by their extensive memory demands and computational burden. One of the significant obstacles to their usage is the extensive usage of floating-point storage and operations in these networks. As a result, there is ample opportunity to compress these networks by adopting a low-bit quantization approach, which can drastically reduce memory and computational requirements. This approach provides a compelling rationale for investigating the development of low-bit quantized SR models [22, 10, 12, 33, 16, 27]. Li *et al.* presented PAMS, which utilizes a trainable truncated parameter to dynamically determine the upper limit of the quantization range for SR models [22]. CADyQ is proposed as a technique designed for SR networks and optimizes the bit allocation for local regions and layers in the input image [10]. Hong *et al.* proposed DAQ, a channel-wise distribution-aware quantization scheme for SR models [12]. Despite these efforts, quantized SR models continue to exhibit a noticeable performance gap in comparison to their full-precision counterparts.

## 3 Method

This section first provides an overview of low-bit quantization for image super-resolution (SR) and points out the limitations of existing low-bit SR networks. We then present our **QuantSR**, an accurate **Quantization** approach for efficient image **Super-Resolution** (see Fig. 2). QuantSR mainly consists of two novel techniques: *Redistribution-driven Learnable Quantizer* (RLQ) and *Depth-dynamic Quantized Architecture* (DQA). RLQ diversifies the information of forward and backward representations. While DQA pushes the accuracy upper bound and achieves dynamic-resource inference. Finally, we discuss how to utilize QuantSR for image SR and optimize the quantized SR network. The implementation details are also provided.

### 3.1 Preliminaries

**SR Network Architecture.** We first outline the basic architecture of quantized SR networks. These networks are designed to take a low-resolution (LR) image denoted as  $I_{LR}$  and produce a corresponding super-resolved image  $I_{SR}$ . The process can be expressed as follows

$$I_{SR} = \mathcal{M}(I_{LR}), \quad (1)$$

where  $\mathcal{M}(\cdot)$  denotes the quantized super-resolution (SR) model. Specifically, the image SR network  $\mathcal{M}(\cdot)$  can be generally divided into three parts: low-level feature extractor  $\mathcal{E}_L(\cdot)$ , high-level feature extractor  $\mathcal{E}_H(\cdot)$ , and reconstruction  $\mathcal{R}(\cdot)$ . The common practice is to apply the network quantization to the high-level feature extractor [22], which costs most of the computational resources among the

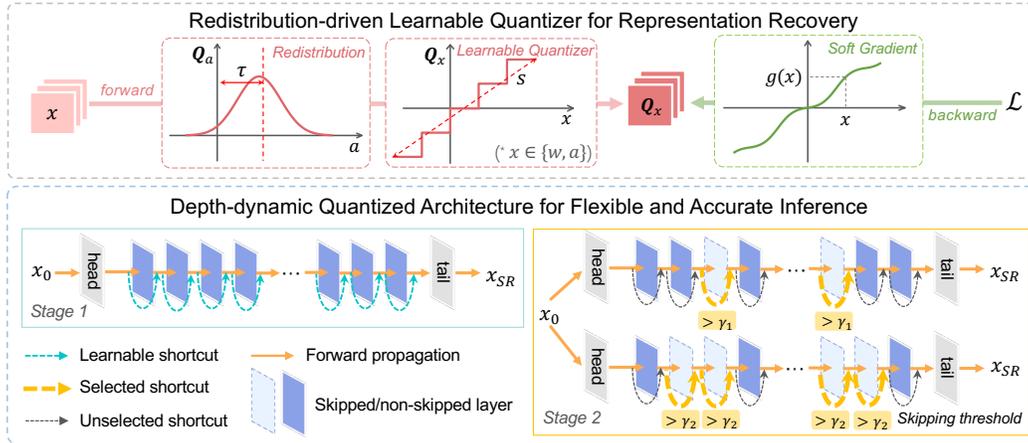


Figure 2: Overview of our QuantSR for image super-resolution network. The upper part is the *Redistribution-driven Learnable Quantizer* (RLQ), where the representation information of quantizers is enhanced in forward and backward propagation of the SR network. The lower one is the *Depth-dynamic Quantized Architecture* (DQA) that breaks the accuracy limitation of quantized SR architecture and allows for the flexible accuracy-efficiency trade-off in inference.

whole SR network. Thus, the image SR process in Eq. (1) can be further expressed as follows

$$I_{SR} = \mathcal{M}(I_{LR}) = \mathcal{R} \circ \mathcal{E}_H \circ \mathcal{E}_L(I_{LR}), \quad (2)$$

where  $\mathcal{E}_H$  denotes the high-level feature extractor and  $\circ$  denotes the connection among network parts.

**Quantization Framework.** For the quantized SR network, the weight  $w$  and activation  $a$  of computing units (such as convolution, linear, and matmul layers) are compressed to low bit-widths by the quantizers  $Q_w(w)$  and  $Q_a(a)$ , respectively. As the common practice [35], these parameters are compressed to lower  $b$  bit-widths by following quantizer  $Q$  with a symmetric mode:

$$Q^b(x) = \text{round} \left( \frac{\text{clip}(x)}{v_b} \right) v_b, \quad (3)$$

where  $x$  denotes the weight  $w$  or activation  $a$ , and  $Q^b(\cdot)$  denotes their quantizer. The function  $\text{clip}(\cdot) = \max(\min(x, a), -a)$  is to limit the range of the inputs, where  $a$  represents the maximum of the absolute value of  $x$ . And  $v_b$  is the map function that scales the higher precision inputs to their lower bit reflections as  $v_b = \frac{a}{2^{b-1}-1}$ . Since the direct differentiation of the discrete quantizer causes all-zero gradients and hinders the backward propagation, straight-through estimation (STE) is used to approximate the gradient of parameters:

$$\frac{\partial Q^b(x)}{\partial x} = \begin{cases} 1 & \text{if } x \in (-a, a) \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

After quantizing the SR networks, the storage size and computation load can be significantly reduced due to the low bit-width and efficient integer operations.

## 3.2 Redistribution-driven Learnable Quantizer

### 3.2.1 Quantization-induced Representation Degradation

While quantization promises less storage and faster inference, it also causes significant performance degradation of compressed SR models. The intuitive reason is that the discretization brought by the quantizer hinders the information of representations in both forward and backward propagations. In forward propagation, the information of quantized parameters is significantly reduced due to the limitation of low bit-width, especially in ultra-low bit quantization. For example, in 2-bit quantization, the number of candidates for a single element drops from  $2^{32}$  in floating point form to only 4. Furthermore, quantizers such as Eq. (3) also lead to homogeneous quantization mappings throughout the SR network. In backward propagation, since the gradient of quantizers cannot be used directly, the gradient obtained by estimators cannot completely reflect the effect of quantizers. This phenomenon results in an information mismatch always between backward and forward propagation, with the former consistently lagging behind. Therefore, the quantizer should be improved to achieve more accurate forward and backward processes, enhancing the representations of the quantized SR network.

### 3.2.2 Redistribution-driven Learnable Quantizer for Representation Recovery

In this work, we proposed a novel Redistribution-driven Learnable Quantizer (RLQ) for our QuantSR (Fig. 3). RLQ is driven by parameter redistribution designs, which introduce learnable parameters into the quantizer to diversify the forward quantization mapping in the whole quantized SR model. The transformation function is added in each quantization interval to enrich the gradient information in the backward. Our RLQ can be expressed as

$$Q_{\text{RLQ}}^b(\mathbf{x}, \hat{v}_b, \hat{\tau}) = \text{round} \left( \phi \left( \frac{\text{clip}(\mathbf{x} + \hat{\tau})}{\hat{v}_b} \right) \right) \hat{v}_b, \quad \phi(\mathbf{x}) = \frac{\tanh(2(\mathbf{x} - \lfloor \mathbf{x} \rfloor) - 1)}{\tanh 1} + \lfloor \mathbf{x} \rfloor + 2^{-1}, \quad (5)$$

where the  $\hat{v}_b$  and  $\hat{\tau}$  denote the learnable interval and mean-shifting parameters, respectively, which are initialized as  $\frac{a}{2^{b-1}-1}$  and 0. The function  $\phi(\cdot)$  is embedded within each quantization interval as a transformation function. This function does not alter the rounded value but rather reduces the gradient of elements that are distant from the interval center. The aim of this function is to incorporate the information reflecting the quantizer's actual behavior while maintaining optimization stability.

The derivative *w.r.t.* the input and learnable parameters used in the backward pass are

$$\frac{\partial Q_{\text{RLQ}}^b(\mathbf{x}, \hat{v}_b, \hat{\tau})}{\partial \mathbf{x}} = \begin{cases} \frac{\partial \phi(\mathbf{x} + \hat{\tau})}{\partial \mathbf{x}} & \text{if } \mathbf{x} \in (-a, a) \\ 0 & \text{otherwise} \end{cases}, \quad \frac{\partial Q_{\text{RLQ}}^b(\mathbf{x}, \hat{v}_b, \hat{\tau})}{\partial \hat{\tau}} = 1 + \frac{\partial \phi(\mathbf{x} + \hat{\tau})}{\partial \hat{\tau}}, \quad (6)$$

$$\frac{\partial Q_{\text{RLQ}}^b(\mathbf{x}, \hat{v}_b, \hat{\tau})}{\partial \hat{v}_b} = \begin{cases} \text{round} \left( \frac{\mathbf{x} + \hat{\tau}}{\hat{v}_b} \right) + \frac{\partial \phi((\mathbf{x} + \hat{\tau}) \hat{v}_b^{-1})}{\partial \hat{v}_b} & \text{if } \mathbf{x} \in (-a, a) \\ -a \text{ or } a & \text{otherwise} \end{cases}.$$

Our proposed  $Q_{\text{RLQ}}^b(\mathbf{x}, \hat{v}_b, \hat{\tau})$  quantizer is embedded in the entire quantized SR network for quantization-aware training to compress weights and activations to low bit-width.

After applying RLQ, both the forward and backward representations of QuantSR are greatly enhanced. (1) For the forward propagation, since introducing the learnable mean-shifting and quantization scale parameters to RLQ, the quantizers are gradually diversified during training. The learnable parameters of the quantizers in the entire network diversify continuously throughout the training process, which significantly reduces the impact of parameter representation degradation caused by parameter discretization. (2) For the backward propagation, the embedded transformation function  $\phi(\cdot)$  can introduce information in the gradient to help accurate updates. Compared with using STE directly, RLQ provides additional gradient-guided information, that is, in each quantization interval, the farther away from the center of the region, the smaller the gradient. The transformation function is embedded in the round function and can be merged at inference time without additional burden.

### 3.3 Depth-dynamic Quantized Architecture

#### 3.3.1 Latent Architecture Bounds for Quantized SR Network

Quantization is a highly effective compression technique that employs low-bit parameters and efficient bitwise operations, making it ideal for fast inference on resource-constrained devices. One of the key benefits of quantization is that it can be applied to SR models without affecting their underlying architecture. However, in our investigation, we observed that the quantized SR network's accuracy is typically limited by that of its full-precision counterpart and that quantization inevitably leads to some loss in accuracy. Furthermore, since SR models come in various sizes to meet different accuracy requirements, striking a balance between accuracy and efficiency becomes critical in practical applications. To address this challenge, we propose a dynamic depth quantization architecture that allows for immediate and adaptive accuracy-efficiency trade-offs in SR applications. This architecture also generates quantization models with greater accuracy at the same depth.

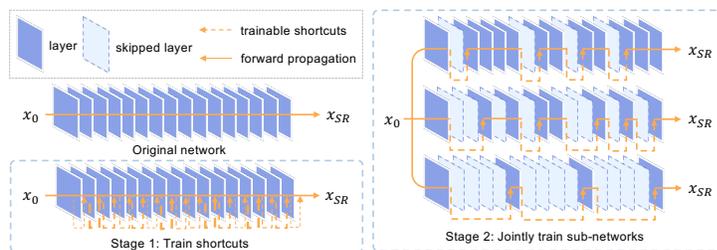


Figure 4: The architecture and training pipeline of DQA, which uses a two-stage strategy to train weight-shared variants with different depths.

However, in our investigation, we observed that the quantized SR network's accuracy is typically limited by that of its full-precision counterpart and that quantization inevitably leads to some loss in accuracy. Furthermore, since SR models come in various sizes to meet different accuracy requirements, striking a balance between accuracy and efficiency becomes critical in practical applications. To address this challenge, we propose a dynamic depth quantization architecture that allows for immediate and adaptive accuracy-efficiency trade-offs in SR applications. This architecture also generates quantization models with greater accuracy at the same depth.

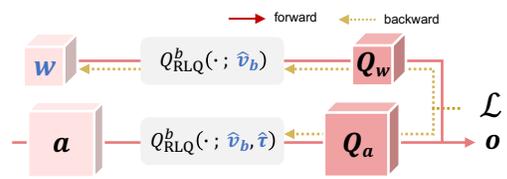


Figure 3: Forward and backward propagation of RLQ. Blue notations are learnable parameters.

### 3.3.2 Depth-dynamic Quantized Architecture for Flexible and Accurate Inference

We propose a Depth-dynamic Quantized Architecture (DQA) for the quantized high-level feature extractor  $\mathcal{E}'_H$  (Fig. 4), which selects a more lightweight model with fewer layers at runtime, directly reducing the computational consumption. We first build the main quantized architecture, which is computationally expensive with learnable short connections. The main architecture  $\mathcal{E}_H^{\text{DQA}}$  containing  $2N$  blocks ( $N \in \{2^n, n \in \mathbf{Z}^+\}$ ), where the initialized  $i$ -th block  $\hat{\Phi}_i^{\text{DQA}}$  is expressed as

$$\hat{\Phi}_i^{\text{DQA}}(\mathbf{x}_i) = \varphi_i(\mathbf{x}_i) + \alpha_i \mathbf{x}_i, \quad (7)$$

where  $\mathbf{x}_i$  denotes the input feature of  $i$ -th block,  $\varphi(\cdot)$  denotes the quantized feature extractor consisting of convolution and activation layers, and  $\alpha$  denotes the learnable scaling of the skip connection.

The training strategy of our architecture is divided into two stages. The **first stage** is a warm-up, which takes 1/5 of the training iterations. In this stage, the entire structure is optimized according to the original loss, and the skip connection is updated during the training process. In the **next stage**, the architecture is derived into a dynamic-depth version. Different derivatives are jointly trained. Specifically, the dynamic-depth architecture is allowed to be derived into different variants with the 100%, 50%, and 25% number of blocks. The variants of DQA can be expressed as

$$\mathcal{E}_H^{\text{DQA}}(x, \text{var}) = \prod_{i=1}^n \Phi_i^{\text{DQA}}(\mathbf{x}_i, \text{var}) = \prod_{i=1}^n (b_i(\text{var})\varphi_i(\mathbf{x}_i) + \alpha_i \mathbf{x}_i), \quad (8)$$

where  $\prod$  denotes composition,  $\text{var} \in (0, 100\%]$  and  $b_i(\text{var}) \in \{0, 1\}$ . The coefficient  $b_i(\text{var})$  of  $i$ -th block is determined based on the value of its scaling  $\alpha_i$  of each block. The  $b[\text{var}]$  is determined based on the value of its short connection  $\alpha$  in each block.

For a specific variant  $\text{var}$ , we select the corresponding proportion of blocks with the smallest short connection values (meaning that the block is less likely to be skipped) after warming up and set their coefficient  $b[\text{var}]$  to 1. The coefficients corresponding to other blocks are set to 0 and their feature extraction processes are skipped. The reserved short connections can also be merged into the previous blocks to achieve acceleration during inference.

### 3.4 Training Pipeline of QuantSR

To optimize the SR model quantized by the proposed QuantSR, we adopt the joint training strategy to update multiple DQA variants, where the RLQ is applied to quantized weight and activation. On the given training dataset  $D = \{I_{\text{LR}}^i, I_{\text{HR}}^i\}_{i=1}^K$  with  $K$  low-resolution inputs  $I_{\text{LR}}$  and corresponding high-resolution ground-truth  $I_{\text{HR}}$ , for a specific variant  $\text{var}$  of quantized SR model  $\mathcal{M}$ , the conventional pixel-wise  $\mathcal{L}_{\text{var}}^{\text{PIX}}$  loss can be expressed as

$$\mathcal{L}_{\text{var}}^{\text{PIX}} = \frac{1}{n} \sum_{i=1}^k \|I_{\text{HR}}^i - \mathcal{M}(I_{\text{LR}}^i, \text{var})\|_{\ell_1}. \quad (9)$$

And the losses of different variants of DQA are accumulated and used to jointly update the network:

$$\mathcal{L}_{\text{tot}} = \sum_{\text{var} \in \text{var}} \frac{\mathcal{L}_{\text{var}}^{\text{PIX}}}{|\text{var}|_{\ell_0}}, \quad \text{var} = \{100\%, 50\%, 25\%\}, \quad (10)$$

where  $|\cdot|_{\ell_0}$  denotes  $\ell_0$  normalization. With the well-designed quantizer, architecture, and training pipeline, our QuantSR enjoys accuracy and efficiency with CNN- and Transformer-based backbones.

## 4 Experimental Results

### 4.1 Settings

**Dataset.** We adhere to the standard procedure in image SR, training on DIV2K [32] and evaluating on Set5 [2], Set14 [37], B100 [29], Urban100 [13], and Manga109 [30].

**Evaluation.** We report the reconstruction performance measured by PSNR and SSIM [34] on the Y channel of the YCbCr space. And we also evaluate the amounts of parameters and computation for the quantized SR model and the full-precision one.

**Proposed Quantization Baselines.** We quantize image SR models with our proposed method, including the CNN- and Transformer-based architectures. For CNN-based SR models, we follow PAMS [22] and CADyQ [10], and use SRResNet as the backbone. For Transformer-based SR models,

Method	#Bit ( $w/a$ )	#Blk	Set5		Set14		B100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRResNet	32/32	16	38.00	0.9605	33.59	0.9171	32.19	0.8997	32.11	0.9282	38.56	0.9770
DoReFa	4/4	16	37.60	0.9589	33.10	0.9133	31.87	0.8954	30.87	0.9151	37.65	0.9746
RLQ	4/4	16	37.72	0.9594	33.25	0.9147	31.98	0.8971	31.15	0.9187	37.99	0.9755
DAQ	4/4	32	37.77	0.9596	33.25	0.9150	32.00	0.8973	31.30	0.9200	38.12	0.9758
		16	37.69	0.9593	33.15	0.9140	31.92	0.8963	30.65	0.9163	37.80	0.9751
		8	37.51	0.9584	32.99	0.9122	31.78	0.8943	30.50	0.9111	37.38	0.9736
QuantSR	4/4	32	37.95	0.9603	33.59	0.9177	32.17	0.8996	31.98	0.9274	38.62	0.9771
		16	37.88	0.9600	33.45	0.9166	32.10	0.8988	31.72	0.9249	38.37	0.9766
		8	37.74	0.9595	33.25	0.9150	31.99	0.8973	31.27	0.9199	37.98	0.9757

Table 1: Ablation study ( $\times 2$  scale) about our proposed Redistribution-driven Learnable Quantizer for Representation Recovery (RLQ) and Depth-dynamic Quantized Architecture (DQA) for flexible and accurate inference. ‘ $w/a$ ’ denotes the weight/activation bits. ‘#Blk’ means residual block number.

we quantize lightweight SwinIR [25]. The implementations of comparison methods follow the official codes [11, 23] and are trained with the same settings as ours. We quantize both the weight and activation of the body part in all the models with low bit-width (*e.g.*, 2, 4, or 8). We denote  $w$ -bit weight and  $a$ -bit activation as  $w/a$ . To ensure a fair comparison, QuantSR reports the performance of the same-depth variant to other networks (*e.g.*, 16 blocks for SRResNet) in the comparison. The complete efficiency performance (32, 16, and 8 blocks) is demonstrated in Sec. 4.3.

**Training Strategy.** In our training process, we follow the practices of previous studies [26, 40, 36, 25] by conducting data augmentation, which involves random rotations of  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ , and horizontal flipping. The models are trained for 300K iterations, with each training batch consisting of 32 image patches. The input size of each patch is  $64 \times 64$ . To optimize our model, we utilize the Adam optimizer [19]. The learning rate is initially set to  $2 \times 10^{-4}$  and is then halved at the 250K-th iteration. All experiments are conducted on NVIDIA RTX A6000 GPUs with PyTorch [31].

## 4.2 Ablation Study

To showcase the efficacy of the techniques employed in our QuantSR, we conduct ablation studies on RLQ and DQA. We employ SRResNet [20] as the image SR backbone and trained it for 200K iterations with a 4-bit setting. To establish a quantization baseline, we used the vanilla quantization method, DoReFa [44]. Subsequently, we incorporate RLQ and/or DQA into SRResNet and binarized it. The results of our experiments are presented in Tab. 1, where we report the PSNR/SSIM values on five test benchmarks. Additionally, we provide detailed analyses of all variants (*e.g.*, 32, 16, and 8 blocks) for the depth-dynamic DQA and QuantSR.

**Redistribution-driven Learnable Quantizer.** The vanilla version of the quantized SR model, DoReFa [44], has demonstrated basic SR performance. However, we have improved upon this by incorporating RLQ for activations, which allows us to learn the shift and numerical scale. Our proposed RLQ has significantly enhanced the performance of the quantized network while reducing the performance drop. In Tab. 1, it is clear that our RLQ achieves remarkable improvements in PSNR and SSIM with little additional computation overhead, particularly with the 4-bit setting, where we achieve around 0.12-0.28 dB and 0.0005-0.0036 improvements, respectively. Our enhanced quantizer significantly improves the representation of information during both forward and backward propagation, leading to improved performance in quantized SR networks. This enhancement results in enhanced representation capabilities and more precise optimization.

**Depth-dynamic Quantized Architecture.** The proposed dynamic quantization algorithm (DQA) can adaptively operate on different versions of a model, based on the energy budget of real-world application scenarios. As shown in Tab. 1, we can use the proposed optimization scheme to jointly train models with varying degrees of quantization, where the variable in Eq. (10) can take values of 100%, 50%, and 25%. The 16-block DQA has achieved an improvement of around 0.05 dB and 0.0010 over the baseline, surpassing the upper limit of accuracy achievable by full-precision models. Notably, the DQA can effectively reduce computational complexity and memory usage by selectively ranking and skipping blocks based on a percentage threshold. Consequently, the 8-block model with the 4/4 setting has the smallest computational and storage footprint. The 32-block variant exhibits a noteworthy accuracy improvement at the cost of increased computational overhead, *e.g.*, the improvement in Set5 is 0.17 dB and 0.0005. The accuracy results of QuantSR also demonstrate that employing both RLQ and DQA techniques can enhance the performance jointly and even elevate it to full precision levels, accomplished by utilizing an efficient 4-bit width.

Method	Scale	#Bit ( $w/a$ )	Set5		Set14		B100		Urban100		Manga109	
			PSNR	SSIM								
Bicubic	$\times 2$	-/-	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	30.80	0.9339
SRResNet [20]	$\times 2$	32/32	38.00	0.9605	33.59	0.9171	32.19	0.8997	32.11	0.9282	38.56	0.9770
SwinIR_S [25]	$\times 2$	32/32	38.14	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.12	0.9783
DoReFa [44]	$\times 2$	8/8	37.32	0.9520	32.90	0.8680	31.69	0.8504	30.32	0.8800	37.01	0.9450
CADyQ [10]	$\times 2$	8/8	37.79	0.9590	33.37	0.9150	32.02	0.8980	31.53	0.9230	38.06	0.9760
DoReFa [44]	$\times 2$	4/4	37.31	0.9510	32.48	0.9091	31.64	0.8901	30.18	0.8780	36.95	0.9440
PAMS [22]	$\times 2$	4/4	37.67	0.9588	33.19	0.9146	31.90	0.8966	31.10	0.9194	37.62	0.9400
CADyQ [10]	$\times 2$	4/4	37.58	0.9580	33.14	0.9140	31.87	0.8960	30.94	0.9170	37.31	0.9740
QuantSR-C (ours)	$\times 2$	4/4	<b>37.80</b>	<b>0.9597</b>	<b>33.35</b>	<b>0.9158</b>	<b>32.04</b>	<b>0.8979</b>	<b>31.46</b>	<b>0.9221</b>	<b>38.25</b>	<b>0.9762</b>
QuantSR-T (ours)	$\times 2$	4/4	<b>38.10</b>	<b>0.9604</b>	<b>33.65</b>	<b>0.9186</b>	<b>32.21</b>	<b>0.8998</b>	<b>32.20</b>	<b>0.9295</b>	<b>38.85</b>	<b>0.9774</b>
DoReFa [44]	$\times 2$	2/2	36.91	0.9470	32.55	0.9071	31.41	0.8868	29.60	0.8740	36.132	0.9410
PAMS [22]	$\times 2$	2/2	34.04	0.8270	30.91	0.8751	30.11	0.8592	27.57	0.8400	31.79	0.9110
CADyQ [10]	$\times 2$	2/2	19.44	0.5610	18.51	0.4810	19.70	0.4760	17.97	0.4550	17.346	0.5830
QuantSR-C (ours)	$\times 2$	2/2	<b>37.57</b>	<b>0.9589</b>	<b>33.09</b>	<b>0.9136</b>	<b>31.84</b>	<b>0.8954</b>	<b>30.77</b>	<b>0.9149</b>	<b>37.60</b>	<b>0.9745</b>
QuantSR-T (ours)	$\times 2$	2/2	<b>37.55</b>	<b>0.9587</b>	<b>33.12</b>	<b>0.9143</b>	<b>31.89</b>	<b>0.8958</b>	<b>30.96</b>	<b>0.9172</b>	<b>37.61</b>	<b>0.9745</b>
Bicubic	$\times 4$	-/-	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SRResNet [20]	$\times 4$	32/32	32.16	0.8951	28.60	0.7822	27.58	0.7364	26.11	0.7870	30.46	0.9089
SwinIR_S [25]	$\times 4$	32/32	32.44	0.8976	28.77	0.7858	27.69	0.7406	26.47	0.7980	30.92	0.9151
DoReFa [44]	$\times 4$	4/4	29.57	0.8369	26.82	0.7352	26.47	0.6971	23.75	0.6898	27.89	0.8634
PAMS [22]	$\times 4$	4/4	31.59	0.8851	28.20	0.7725	27.32	0.7220	25.32	0.7624	28.86	0.8805
CADyQ [10]	$\times 4$	4/4	31.48	0.8830	28.05	0.7690	27.21	0.7240	25.09	0.7520	28.82	0.8840
QuantSR-C (ours)	$\times 4$	4/4	<b>32.00</b>	<b>0.8924</b>	<b>28.50</b>	<b>0.7799</b>	<b>27.52</b>	<b>0.7342</b>	<b>25.88</b>	<b>0.7807</b>	<b>30.15</b>	<b>0.9040</b>
QuantSR-T (ours)	$\times 4$	4/4	<b>32.18</b>	<b>0.8941</b>	<b>28.63</b>	<b>0.7822</b>	<b>27.59</b>	<b>0.7367</b>	<b>26.11</b>	<b>0.7871</b>	<b>30.49</b>	<b>0.9087</b>
DoReFa [44]	$\times 4$	2/2	30.54	0.8610	27.50	0.7538	26.90	0.7098	24.44	0.7242	27.31	0.8502
PAMS [22]	$\times 4$	2/2	29.20	0.8239	26.61	0.7273	26.36	0.6934	23.58	0.6812	25.59	0.8012
CADyQ [10]	$\times 4$	2/2	19.67	0.5380	19.30	0.4740	19.80	0.4620	17.97	0.4360	17.30	0.5640
QuantSR-C (ours)	$\times 4$	2/2	<b>31.30</b>	<b>0.8819</b>	<b>28.08</b>	<b>0.7694</b>	<b>27.23</b>	<b>0.7246</b>	<b>25.13</b>	<b>0.7537</b>	<b>28.81</b>	<b>0.8844</b>
QuantSR-T (ours)	$\times 4$	2/2	<b>31.53</b>	<b>0.8845</b>	<b>28.16</b>	<b>0.7715</b>	<b>27.28</b>	<b>0.7274</b>	<b>25.26</b>	<b>0.7609</b>	<b>29.06</b>	<b>0.8898</b>

Table 2: Quantitative results. SRResNet and SwinIR-S are used as full-precision backbones. ‘ $w/a$ ’ denotes the weight/activation bits. The best and second best results are colored with red and cyan.

### 4.3 Image Super-Resolution (SR)

We select SRResNet [20] (*i.e.*, 1,367K ( $\times 2$ ) and 1,515K ( $\times 4$ ) Params) and SwinIR\_S [25] (*i.e.*, lightweight SwinIR with 910K ( $\times 2$ ) and 930K ( $\times 4$ ) Params) as CNN and Transformer backbones, respectively. The corresponding QuantSR variants are QuantSR-C and QuantSR-T, respectively. And we then compare with recent quantization methods, *e.g.*, DoReFa [44], PAMS [22], and CADyQ [10].

**Quantitative Results.** In Tab. 2, we provide PSNR and SSIM. Our 4-bit QuantSR-C achieves comparable or superior PSNR/SSIM scores than 8-bit DoReFa and CADyQ with scale  $\times 2$ . In the 4-bit case, our QuantSR-C achieves 0.52 dB/0.0051 ( $\times 2$ ) and 0.79 dB/0.0287 ( $\times 4$ ) higher PSNR/SSIM values than CADyQ [10] on Urban100. The case of the Transformer is more challenging, as we observe a larger performance gap between the quantized QuantSR-T and the full-precision model compared to the quantized SRResNet. However, our QuantSR effectively narrows this gap, with a constant improvement compared to previous techniques, particularly at lower bit widths. For instance, with a 2-bit setting on SRResNet, our QuantSR-C outperforms PAMS by a significant 3.20 dB/0.0749 ( $\times 2$ ) and 1.55 dB/0.0725 ( $\times 4$ ) on Urban100. It also is significantly higher than the performance obtained with the SOTA CADyQ and approaches that of the full-precision model. Similarly, on Urban100, our QuantSR-T improves 1.26 dB/0.0125 ( $\times 2$ ) over CADyQ in the 4-bit setting.

**Compression Ratio.** By utilizing a combination of quantization and dynamic, lightweight architecture, compression ratio of image SR models is significantly increased. In Tab. 3, we present the compression ratio and speedup in terms of Params and Ops, respectively. By quantizing the

Method	#Bit ( $w/a$ )	#Blk	Params (K)	Ops (G)	Urban100	
			( $\downarrow$ Ratio)	( $\downarrow$ Ratio)	PSNR	SSIM
SRResNet	32/32	16	1,367 (0%)	90.1 (0%)	32.16	0.8951
QuantSR-C	4/4	32	451 ( $\downarrow$ 67.0%)	29.9 ( $\downarrow$ 66.9%)	32.17	0.8943
		16	303 ( $\downarrow$ 77.8%)	20.2 ( $\downarrow$ 77.5%)	32.00	0.8924
		8	230 ( $\downarrow$ 83.1%)	15.4 ( $\downarrow$ 82.9%)	31.75	0.8894
QuantSR-C	2/2	32	170 ( $\downarrow$ 87.6%)	11.5 ( $\downarrow$ 87.2%)	31.48	0.8849
		16	161 ( $\downarrow$ 88.2%)	10.9 ( $\downarrow$ 87.9%)	31.30	0.8819
		8	156 ( $\downarrow$ 88.6%)	10.6 ( $\downarrow$ 88.3%)	31.04	0.8771

Table 3: Compression ratio of 2-bit and 4-bit SRResNet ( $\times 2$ ), and their input sizes are  $3 \times 256 \times 256$  for calculating Ops.

full-precision SRResNet to 2 and 4 bit-widths, we can reduce both the model size (*i.e.*, Params) and operations (*i.e.*, Ops) considerably. Following PAMS [22], we only quantize the weights and activations in the high-level feature extractor part, but we calculate the compression ratio and speedup for the entire model. With a 4-bit setting, our QuantSR-C achieves approximately 77.8% and 77.5% compression ratio for parameters and operations, respectively, through quantization alone. By integrating DQA for the 8-block variant, we were able to push the compression ratios to 83.1% and 82.9%, respectively. Furthermore, with a 2-bit setting, the operation compression ratios of 16- and 8-block variants can even reach 87.9% and 88.3%, respectively. Regarding the variant with 32

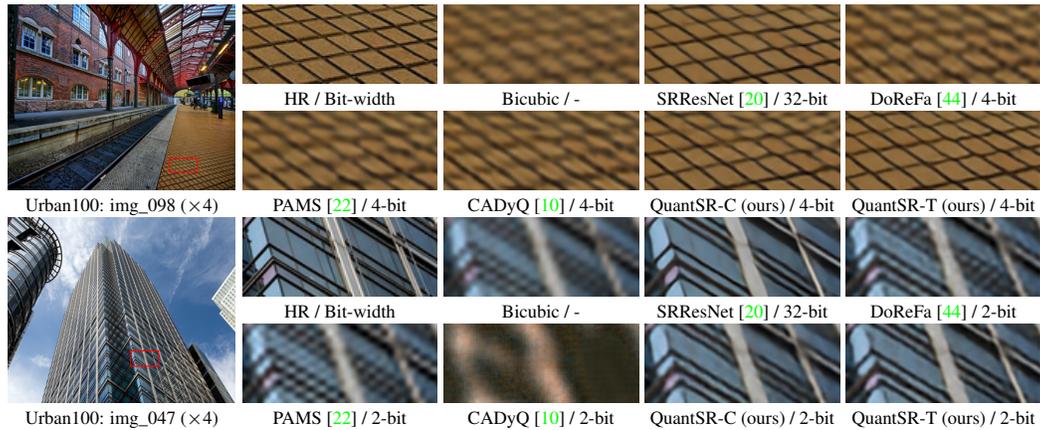


Figure 5: Visual comparison ( $\times 4$ ) with lightweight SR in terms of 4-bit and 2-bit.

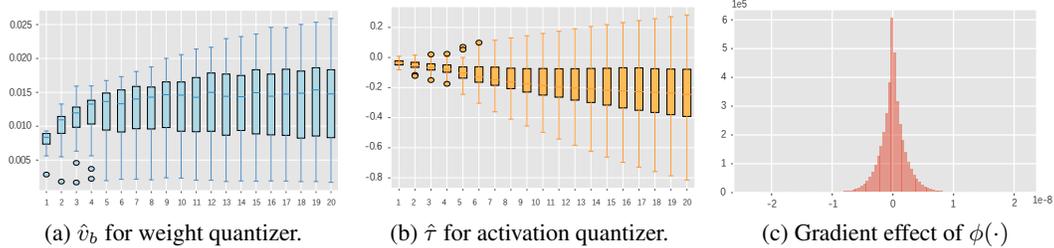


Figure 6: The statistics of RLQ demonstrate its representation improvement in forward and backward.

blocks, 4-bit QuantSR-C model also achieves notable parameter and operation savings of 67.0% and 66.9%, respectively, while obtaining comparable accuracy with the 16-block full-precision model.

#### 4.4 Visualization

**Visual Results.** In Fig. 5, we provide visual results of representative methods with scale  $\times 4$  in terms of 2-bit and 4-bit cases. For each case, we compare with several quantization methods, like DoReFa [44], PAMS [22], and CADyQ [10]. Our QuantSR-C and QuantSR-T recover more structural details and alleviate more blurring artifacts than other quantization methods. Consequently, the visual difference between our QuantSR and the full-precision one is small. These visual comparisons further demonstrate the effectiveness of our QuantSR, which is consistent with the observations in Tab. 2.

**Training Statistics.** In Fig. 6, we present statistics of the learnable parameters and function effects in our RLQ for every 100 iterations of training. Specifically, we report the statistics of  $\hat{v}_b$ , which represents the activations,  $\hat{\tau}$ , which represents the weights, and the gradient effects of  $\phi(\cdot)$ . Figures 6 (a) and (b) illustrate that the learnable parameters are initially initialized in close proximity to each other. However, as training progresses, the optimized parameters diverge in various directions. This indicates that the quantizers containing these parameters in QuantSR are becoming more diverse, thus recovering the lost representation information due to discretization. In Fig. 6 (c), we depict the gradient change induced by a specific embedded transformation function  $\phi(\cdot)$  during backpropagation. While the forward results and training stability remain unaffected, the gradient guidance information provided by this function is significant. Consequently, it promotes the optimization of QuantSR, leading to an overall improvement in the results.

## 5 Conclusion

In this paper, we propose QuantSR, a novel quantized image super-resolution (SR) network that overcomes the limitations of existing low-bit quantized SR models. The proposed network employs a Redistribution-driven Learnable Quantizer (RLQ) to improve the representation ability of quantized networks and a Depth-dynamic Quantized Architecture (DQA) to achieve flexible inference and break the upper limit of accuracy. The authors demonstrate that QuantSR outperforms existing state-of-the-art quantized SR networks in both accuracy and computational efficiency. Additionally, we provide QuantSR-C and QuantSR-T for both convolution and Transformer versions, respectively, to demonstrate the scheme's satisfactory architecture generality.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China (No. 62022009), the State Key Laboratory of Software Development Environment (SKLSDE-2022ZX-23), and the Huawei Technologies Oy (Finland) Project.

## References

- [1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, 2018. 3
- [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, Yongbing Zhang, Linghe Kong, and Xin Yuan. Cross aggregation transformer for image restoration. In *NeurIPS*, 2022. 1
- [4] 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. 3
- [5] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *ICCVW*, 2019. 1, 3
- [6] Xiangxiang Chu, Bo Zhang, Hailong Ma, Ruijun Xu, and Qingyuan Li. Fast, accurate and lightweight super-resolution with neural architecture search. *arXiv preprint arXiv:1901.07261*, 2019. 3
- [7] 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. 2
- [8] 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, 3
- [9] Zibin He, Tao Dai, Jian Lu, Yong Jiang, and Shu-Tao Xia. Fakd: Feature-affinity based knowledge distillation for efficient image super-resolution. In *ICIP*, 2020. 3
- [10] 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, 6, 8, 9
- [11] Cheeun Hong, Sungyong Baik, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Cadyq, 05 2023. 7
- [12] 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
- [13] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 6
- [14] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training quantization with small calibration sets. In *ICML*, 2021. 1, 3
- [15] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*, 2019. 3
- [16] Andrey Ignatov, Radu Timofte, Maurizio Denna, Abdel Younes, Ganzorig Gankhuyag, Jingang Huh, Myeong Kyun Kim, Kihwan Yoon, Hyeon-Cheol Moon, Seunggho Lee, et al. Efficient and accurate quantized image super-resolution on mobile npus, mobile ai & aim 2022 challenge: report. In *ECCVW*, 2022. 3
- [17] Divyansh Jhunjhunwala, Advait Gadhikar, Gauri Joshi, and Yonina C Eldar. Adaptive quantization of model updates for communication-efficient federated learning. In *ICASSP*, 2021. 3
- [18] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, 2016. 1
- [19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7
- [20] 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, 7, 8, 9
- [21] Wonkyung Lee, Junghyup Lee, Dohyung Kim, and Bumsub Ham. Learning with privileged information for efficient image super-resolution. In *ECCV*, 2020. 3
- [22] 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, 6, 8, 9

- [23] Huixia Li, Chenqian Yan, Shaohui Lin, Xiawu Zheng, Baochang Zhang, Fan Yang, and Rongrong Ji. Pams, 05 2023. 7
- [24] 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, 3
- [25] Jingyun Liang, Jiezhong Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCVW*, 2021. 1, 7, 8
- [26] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 1, 7
- [27] Jingyu Liu, Qiong Wang, Dunbo Zhang, and Li Shen. Super-resolution model quantized in multi-precision. *Electronics*, 10(17):2176, 2021. 3
- [28] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*, 2018. 2
- [29] 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
- [30] 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
- [31] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS*, 2017. 7
- [32] 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. 6
- [33] Hu Wang, Peng Chen, Bohan Zhuang, and Chunhua Shen. Fully quantized image super-resolution networks. In *ACM MM*, 2021. 3
- [34] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 6
- [35] Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*, 2020. 4
- [36] Jingwei Xin, Nannan Wang, Xinrui Jiang, Jie Li, Heng Huang, and Xinbo Gao. Binarized neural network for single image super resolution. In *ECCV*, 2020. 7
- [37] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Proc. 7th Int. Conf. Curves Surf.*, 2010. 6
- [38] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *CVPR*, 2018. 1
- [39] Jiale Zhang, Yulun Zhang, Jinjin Gu, Yongbing Zhang, Linghe Kong, and Xin Yuan. Accurate image restoration with attention retractable transformer. In *ICLR*, 2023. 1
- [40] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, 2018. 7
- [41] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 1
- [42] Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Aligned structured sparsity learning for efficient image super-resolution. In *NeurIPS*, 2021. 3
- [43] Zixiang Zhao, Shuang Xu, Chunxia Zhang, Junmin Liu, Jianshe Zhang, and Pengfei Li. Didfuse: Deep image decomposition for infrared and visible image fusion. In *IJCAI*, 2020. 1
- [44] 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, 3, 7, 8, 9