CADyQ: Content-Aware Dynamic Quantization for Image Super-Resolution

Cheeun Hong[1], Sungyong Baik[3], Heewon Kim[1],
Seungjun Nah[1,4], and Kyoung Mu Lee[1,2]

[1] Dept. of ECE & ASRI, {cheeun914, ghimhw, kyoungmu}@snu.ac.kr
[2] IPAI, Seoul National University
[3] Dept. of Data Science, Hanyang University dsybaik@hanyang.ac.kr
[4] NVIDIA seungjun.nah@gmail.com

Abstract. Despite breakthrough advances in image super-resolution (SR) with convolutional neural networks (CNNs), SR has yet to enjoy ubiquitous applications due to the high computational complexity of SR networks. Quantization is one of the promising approaches to solve this problem. However, existing methods fail to quantize SR models with a bit-width lower than 8 bits, suffering from severe accuracy loss due to fixed bit-width quantization applied everywhere. In this work, to achieve high average bit-reduction with less accuracy loss, we propose a novel **C**ontent-**A**ware **Dy**namic **Q**uantization (CADyQ) method for SR networks that allocates optimal bits to local regions and layers adaptively based on the local contents of an input image. To this end, a trainable bit selector module is introduced to determine the proper bit-width and quantization level for each layer and a given local image patch. This module is governed by the quantization sensitivity that is estimated by using both the average magnitude of image gradient of the patch and the standard deviation of the input feature of the layer. The proposed quantization pipeline has been tested on various SR networks and evaluated on several standard benchmarks extensively. Significant reduction in computational complexity and the elevated restoration accuracy clearly demonstrate the effectiveness of the proposed CADyQ framework for SR. Codes are available at https://github.com/Cheeun/CADyQ.

1 Introduction

Image super-resolution (SR) is a fundamental low-level vision computer problem that aims to restore the high-resolution (HR) image from its corresponding low-resolution (LR) image. Owing to the remarkable success in deep learning approaches [10,27,34,40,49,50], high-fidelity images could be obtained using state-of-the-art super-resolution networks. Such modern deep learning models, however, rely on advanced architectures with high computational costs, thereby limiting their applications, especially in resource-limited environments.

Quantization is one of the promising approaches for reducing the computational complexity of neural networks. In particular, network quantization has

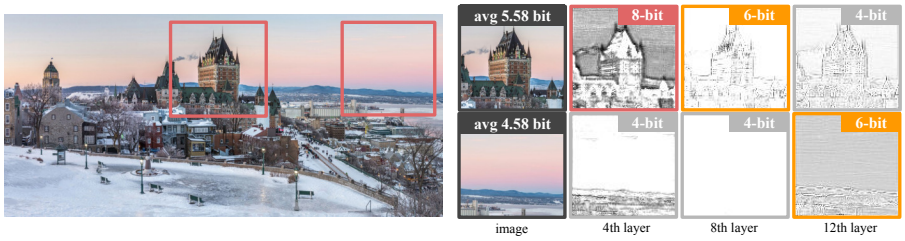
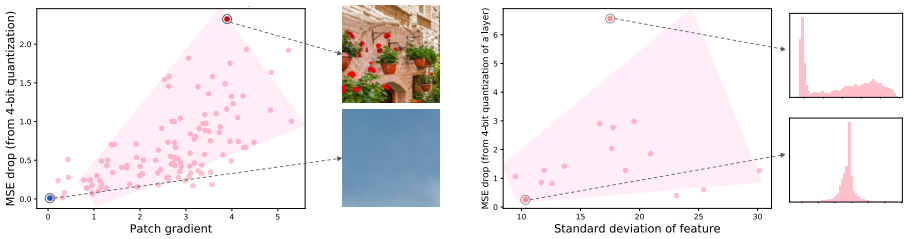


Fig. 1: The dynamic bit-width allocation by CADyQ. Examples from CADyQ applied to a recent SR network, CARN [2]. Our framework demonstrates a dynamic bit-width allocation per patch and layer with a minimal PSNR drop (<0.05 dB). Higher bit-widths are allocated to features containing more structural information or contours

greatly reduced computation loads without a significant accuracy loss, especially for high-level vision tasks (*e.g.*, classification) [7,18,51]. Recently, there have also been attempts to quantize SR networks, either by learning parameters for the binarization of each convolution weight [38] or by learning a quantization range for each layer [31]. However, unlike quantizing high-level vision networks, quantizing SR networks to bit-width lower than 8 bit while maintaining the performance remains a challenging problem [22].

As a key to the above issue, we find that the existing methods do not consider the image structure and locality information, employing a quantized network with fixed bit for all regions of given input images. This leads to processing image regions of less structural information with unnecessarily high bits. In this work, we observe that different local regions (*i.e.*, patches of a certain size) exhibit different amounts of SR performance degradation from quantization, as illustrated in Fig. 2a. In particular, patches with complex structures or contents tend to suffer more from performance degradation than patches with simple contents. Furthermore, we also observe that the quantization sensitivity varies among layers, even for the same patch, as illustrated in Fig. 2b. The observations suggest that different patches and layers require different amounts of computation and thus different bit-widths, providing motivations for a dynamic patch-and-layer-wise bit-width quantization.

Therefore, we propose a new quantization pipeline, dubbed **C**ontent-**A**ware **Dy**namic **Q**uantization (CADyQ), that dynamically selects a quantization bit-width for each convolution layer based on the quantization sensitivity of its input contents (*i.e.*, each patch and layer feature), as demonstrated in Fig. 1. However, the direct measurement of the quantization sensitivity is unsuitable as it requires ground-truth high-resolution images to measure the performance degradation. In order to estimate the quantization sensitivity, the proposed pipeline employs the average gradient magnitude of the input patch and the standard deviation of the layer feature, based on the observed correlation in Fig. 2. Then, we introduce a lightweight bit selector that employs a linear layer conditioned on the estimated



(a) Quantization sensitivity v.s. patch image gradient

(b) Quantization sensitivity v.s. standard deviation of layer features

Fig. 2: The motivation of our framework: the different quantization sensitivity per layer and per patch. Quantization sensitivity is measured with restoration performance (*e.g.*, mean-square error (MSE) between the output image and ground-truth HR image) degradation due to quantization. We observe **(a)** a correlation between the average magnitude of image gradient [13] and the quantization sensitivity of each patch. Patches with complex (simple) structures exhibit high (low) image gradient magnitude and suffer more (less) MSE drop from quantization. Also, we notice a **(b)** strong correlation between feature standard deviation and the quantization sensitivity of each layer feature for the given patch. Layers with high (low) feature standard deviation bring more (less) MSE drop from quantizing the given layer.

quantization sensitivity, to determine the bit-width of the feature for each input patch and layer.

Furthermore, a new regularization loss function is introduced to facilitate the bit-width selection process. The proposed loss function penalizes the bit selector if a high (low) bit is selected for features with a small (large) quantization sensitivity. This leads the bit selector to reserve more computation resources for features that are more critical to the restoration performance, while minimizing the resources for features with less impact on the performance.

The experimental results demonstrate the outstanding performance of the proposed quantization mechanism across various SR networks, underlining the effectiveness and importance of selecting a different bit-width for each patch and layer. Overall, our contributions can be summarized as follows:

- For the first time, we observe that the sensitivity of restoration accuracy to low-bit quantization varies across different local image regions and the SR network layers.
- Accordingly, we present a new quantization framework CADyQ that quantizes SR networks with a different bit-width for each patch and layer, by adding a lightweight bit selector module that is conditioned on the estimated quantization sensitivity.
- A novel regularization loss term is introduced to encourage the proposed framework to find a better balance between the computational complexity and overall restoration performance.

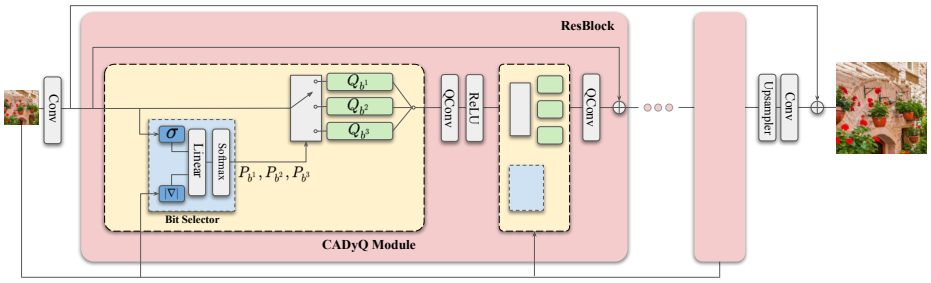


Fig. 3: The overview of the proposed quantization framework CADyQ for SR network, which we illustrate with a residual block based backbone. For each given patch and each layer, our CADyQ module introduces a light-weight bit selector that dynamically selects the bit-width and its corresponding quantization function Q_{b^k} (Eq. (3)) among the candidate quantization functions with distinct bit-widths. The bit selector is conditioned on the estimated quantization sensitivity (the average gradient magnitude $|\nabla|$ of the given patch and the standard deviation σ of the layer feature). Qconv denotes the convolution layer of the quantized features and weights

2 Related Works

Super-Resolution Neural Networks. Convolutional neural network (CNN) based approaches [29,34] have greatly improved the performance of image super-resolution (SR) methods, however, with heavy computational resources. The massive computations of SR networks have limited the application on real-world mobile devices, spurring the recent interest in lightweight SR networks. Since then, new lightweight architectures have been investigated [9,20,21,49] or searched [8,26,32,33,39]. Recently, a few works have introduced adaptive SR networks that aim to achieve efficient inference for a given input [28,36,43,45,48]. These methods mostly focus on reducing the network depth or the number of channels that still rely on heavy floating-point operations, while our focus is to lower the precision of floating-point operations with network quantization.

Neural Network Quantization. Network quantization provides an alternative approach for making networks efficient by mapping 32-bit floating point values of feature maps and weights to lower bit values [5,7,12,25,30,51,52]. Few recent works have attempted to allocate different bit-widths for different layers [6,11,15,24,37,42,47]. However, these approaches target high-level tasks and thus do not consider the distinct local regions that we observe to play a key role in obtaining an efficient network for super-resolution.

Quantized Super-Resolution Models. In contrast to high-level vision tasks, super-resolution poses different challenges due to inherently high accuracy sensitivity to quantization [22,38,46]. A few works have attempted to recover the accuracy by modifying the network architecture [3,23,46]. However, the methods are applicable to specific models and thus not generalizable to other SR architectures. For a general quantization method for SR networks, PAMS [31] learns the quantization intervals of different layers to adapt to vastly distinct distributions

in the features of SR networks (due to the absence of BN layers), and DAQ [17] further achieves ultra-low bit quantization on SR by utilizing different quantization function parameters for each feature channel. Furthermore, Wang *et al.* [41] has proposed quantizing features from all layers and skip connections of SR networks. Considering the varying degree of quantization sensitivity inside the network, Liu *et al.* [35] manually allocated a bit-width for each stage of a network. However, these works apply a fixed bit-width either throughout different input images [35] or both images and network layers [41]. In contrast, we observe that the quantization sensitivity varies throughout the network layers and images. Thus, we propose a new quantization framework that dynamically selects the appropriate bit-width for each layer feature based on its quantization sensitivity of each content (*i.e.*, patch and layer feature).

3 Proposed Method

3.1 Preliminaries

Generally, to replace the majority of floating-point operations with lower-bit operations in CNNs, the input feature and weight of each convolutional layer are respectively quantized [5,7,25]. Given an input feature of the j-th convolutional layer $\boldsymbol{x}^j \in \mathbb{R}^{N\times C\times H\times W}$, where B, C, H, and W denote the mini-batch size, number of channels, height, and width of the feature, a quantization function $Q_b(\cdot)$ quantizes the feature $\boldsymbol{x}^j$ into its low-bit counterpart $\boldsymbol{x}_q^j$ of bit-width b:

$$\boldsymbol{x}_q^j \equiv Q_b(\boldsymbol{x}^j) = \lfloor \text{clamp}(\boldsymbol{x}^j, a) \cdot \frac{s(b)}{a} \rceil \cdot \frac{a}{s(b)}. \tag{1}$$

$\boldsymbol{x}^j$ is first truncated with $\text{clamp}(\cdot, a)$ into the range of $[-a, a]$, and then scaled to $[-1, 1]$ with the scale parameter a. Then, $\boldsymbol{x}^j$ is scaled to the integer range of the given bit-width b, $[-s(b), s(b)]$ where $s(b)=2^{b-1}-1$. Consequently, the features in integer range are then rounded to integer values with $\lfloor \cdot \rceil$, and then rescaled back to range $[-a, a]$. For quantizing features of SR networks, scale parameter a is either implemented with a learnable parameter [31] or a moving average of batch-wise max values [41]. At the output of the ReLU layers, since the values are non-negative, the output values are truncated into the range of $[0, a]$ and then scaled to integer range $[0, s(b)]$ with $s(b)=2^b-1$. Similarly, a weight of the j-th convolutional layer $\boldsymbol{w}^j \in \mathbb{R}^{C\times C_{out}\times F\times F}$ is quantized to $\boldsymbol{w}_q^j$:

$$\boldsymbol{w}_q^j \equiv Q_b(\boldsymbol{w}^j) = \lfloor \text{clamp}(\boldsymbol{w}^j, a_w^j) \cdot \frac{s(b)}{a_w^j} \rceil \cdot \frac{a_w^j}{s(b)}, \tag{2}$$

where C and C_{out} are the number of input and output channels, F is the kernel size of the convolution filter, and a_w^j is the scale parameter for the corresponding weight $\boldsymbol{w}^j$. In quantization for SR networks, the weight scale parameter a_w^j is often determined simply by $a_w^j = \max(|\boldsymbol{w}^j|)$ [31].

3.2 Motivation

Previous SR quantization works [31,41] have quantized the network with a fixed bit-width b, as formulated in Eq. (1). However, our observations in Fig. 2 hint the disadvantages of a fixed bit-width quantization in SR. In particular, different patches and network layers exhibit different degrees of quantization sensitivity (*i.e.*, SR performance drop from a fixed-bit quantization). As such, we aim to dynamically assign bit-widths for the features based on the quantization sensitivity of *contents* (*i.e.*, the input patch *and* the layer), thereby naming our proposed framework **C**ontents-**A**ware **Dy**namic **Q**uantization (CADyQ).

3.3 Proposed Quantization Module (CADyQ)

In our proposed framework, each convolutional layer has a quantization module, which in turn consists of K bit-width quantization function candidates, one of which is selected by a bit selector, as illustrated in Fig. 3.

Dynamic Feature Quantization. To dynamically quantize features with a different bit-width for each i-th patch and j-th layer, a single quantization function will be selected in the CADyQ module among K number of candidate quantization functions with distinct bit-widths. Each quantization function $Q_{b^k_{i,j}}(\cdot)$ of bit-width $b^k_{i,j}$ (k=1, ..., K) will, when selected, quantize the feature of the i-th patch and j-th layer $\boldsymbol{x}^j_i$ with

$$Q_{b^k_{i,j}}(\boldsymbol{x}^j_i) = \lfloor \text{clamp}(\boldsymbol{x}^j_i, a_k) \cdot \frac{s(b^k_{i,j})}{a_k} \rceil \cdot \frac{a_k}{s(b^k_{i,j})}, \tag{3}$$

where $s(b_k)$=2^{b_k-1}-1 is the integer range of the bit-width b_k and a_k is the scale parameter. Once the bit-width is selected to be $b^{k^*}_{i,j}$ for i-th input patch and j-th layer, the resulting quantized counterpart of $\boldsymbol{x}^j_i$ is

$$\boldsymbol{x}^j_{i,q} \equiv Q_{b^{k^*}_{i,j}}(\boldsymbol{x}^j_i),$$

where $Q_{b^{k^*}_{i,j}}$ is the quantization function for $\boldsymbol{x}^j_i$, corresponding to the selected bit-width $b^{k^*}_{i,j}$. Note that we simply use a linear symmetric quantization function and a learnable scale parameter a_k for each quantization function, as in [31].

Bit-Width Selection. To facilitate the bit-width selection, we use a lightweight bit selector that assigns a probability to each bit-width. Then, the bit-width with the highest probability is selected:

$$b^{k^*}_{i,j} = \begin{cases} \arg\max_{b^k_{i,j}} P_{b^k_{i,j}}(\boldsymbol{x}^j_i) & \text{forward,} \\ \sum_{k=1}^{K} b^k_{i,j} \cdot P_{b^k_{i,j}}(\boldsymbol{x}^j_i) & \text{backward,} \end{cases} \tag{4}$$

where $P_{b^k_{i,j}}$ is the probability assigned to the bit-width $b^k_{i,j}$ and its corresponding quantization function and $\sum_k P_{b^k_{i,j}} = 1$. We desire our bit selector network

to predict a high probability to a high bit-width for features that have high quantization sensitivity (high accuracy drop from quantization) and a low bit-width for features with low quantization sensitivity. However, it is infeasible to directly measure the quantization sensitivity of each feature without access to a ground-truth HR patch for the given input LR patch. Therefore, upon the correlations observed in Fig. 2, we estimate the quantization sensitivity for each layer and the given input patch with the average magnitude of image gradient [13] of a patch and the standard deviation of a feature. Conditioned on the average gradient magnitude of a patch and the standard deviation of a feature, our bit selector assigns the probability to each bit-width candidate for $\boldsymbol{x}_i^j$, the feature of the j-th layer and input patch I_i:

$$P_{b_{i,j}^k}(\boldsymbol{x}_i^j) = \frac{\exp(f(\sigma(\boldsymbol{x}_i^j), |\nabla I_i|))}{\sum_{k=1}^{K} \exp(f(\sigma(\boldsymbol{x}_i^j), |\nabla I_i|))}, \tag{5}$$

where $\sigma(\boldsymbol{x}_i^j) \in \mathbb{R}^C$ measures the channel-wise standard deviation and $|\nabla I_i| \in \mathbb{R}^2$ measures the average magnitude of the image gradients from the patch I_i in horizontal and vertical directions [13]. We concatenate the two metrics and then pass it through a fully connected layer $f : \mathbb{R}^{C+2} \rightarrow \mathbb{R}^K$, based on the observed positive correlation between the measured quantization sensitivity and the feature standard deviation or the average gradient magnitude of each patch. While we make observations on the correlation using the layer-wise standard deviation for the clarity, the bit selector is conditioned on the channel-wise standard deviation, which is observed to have more fine-grained information, as discussed in the supplementary document.

Backpropagation. Selecting a quantization function of the max probability, however, is a discrete non-differentiable process and cannot be optimized end-to-end. Hence, we employ the straight-through estimator [4] to make the process differentiable. The discrete bit-width selection is replaced with its differentiable approximation, where each candidate bit-width is weighted by the probability distribution predicted by the bit selector (Eq. (5)), during backpropagation:

$$\boldsymbol{x}_{i,q}^j = \begin{cases} Q_{b_{i,j}^{k^*}}(\boldsymbol{x}_i^j) & \text{forward,} \\ \sum_{k=1}^{K} Q_{b_{i,j}^k}(\boldsymbol{x}_i^j) \cdot P_{b_{i,j}^k}(\boldsymbol{x}_i^j) & \text{backward.} \end{cases}$$

Weight Quantization. Weights are quantized with a fixed bit-width, as in Eq. (2). While weights can be also quantized dynamically, we focus on the dynamic quantization of input features, motivated by the observations of the correlations between the quantization sensitivity and local image contents (*i.e.*, patches and layer features) in Fig. 2.

3.4 Bit Loss

Previous works [31,41] have focused on optimizing the performance of a quantized network with a fixed bit-width, by using a pixel-wise L1 loss and knowledge

distillation loss [16] with the original unquantized network. On the other hand, we aim to find an efficient quantization for the given feature dynamically. Hence, we need a regularization loss term to strike a balance between the restoration performance and the quantization rate. Directed by a similar goal, few neural architecture search (NAS)-based mixed-precision quantization approaches [6,47] utilize bit regularization loss to optimize the computational resources of the quantized network. The typical bit regularization loss penalizes the total number of operations weighted by its bit-width of the currently selected network:

$$\mathcal{L}_b = \sum_{j=1}^{M} \sum_{i=1}^{N} b_{i,j}^{k^*} \cdot \mathrm{OPs}(\boldsymbol{x}_i^j), \tag{6}$$

where N is the batch size; M is the number of quantized layers in the network; $b_{i,j}^{k^*}$ is the selected bit-width corresponding to the feature of i-th patch and j-th layer according to Eq. (4); and $\mathrm{OPs}(\cdot)$ is the number of operations for convoluting the given feature. However, this standard bit regularization loss equally penalizes the quantization modules of different layers when each layer can have a different impact on the overall performance after quantization, as observed in Fig. 2b.

To achieve a better trade-off between the computational cost and restoration performance, the bit-widths of quantization modules with a larger impact on performance should be penalized less than those of the quantization modules with less impact. As a result, the layers with greater impact on the overall performance will have higher bit-width assigned. To this end, we modify the bit regularization loss by weighting each selected bit-width with the probability estimated by our bit selector, which is conditioned on the estimated quantization sensitivity (and thus estimated impact on the overall performance). Given feature $\boldsymbol{x}_i^j$ of the j-th layer for the i-th patch, our weighted bit regularization loss is

$$\mathcal{L}_{wb} = \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{b_{i,j}^{k^*}}{\sum_k b_{i,j}^k \cdot \mathrm{sg}[P_{b_{i,j}^k}(\boldsymbol{x}_i^j)]} \cdot \mathrm{OPs}(\boldsymbol{x}_i^j), \tag{7}$$

where $\mathrm{sg}[\cdot]$ denotes stop gradient operation. Specifically, the denominator represents the expected bit-width during training, while the numerator represents the selected bit-width. When the expected bit-width is smaller than the selected bit-width, it results in a larger regularization term and hence stronger penalization. This penalization enforces a lower bit-width assignment on the feature estimated to be less sensitive to quantization. For example, when the probability distribution from the bit selector network for each bit-width 4, 6, and 8 is [0.1, 0.1, 0.8], the quantization module should be regularized less than [0.2, 0.2, 0.6]. The feature that corresponds to the former probability distribution can be considered to be more vulnerable to the performance drop from quantization. On the other hand, when the expected bit-width is larger than the selected bit-width, a larger expected bit-width is regularized less. This enables the bit selector to select a higher bit-width for more quantization-sensitive features.

Then, our final objective function becomes

$$\mathcal{L} = w_1 \mathcal{L}_1 + w_{\mathrm{reg}} \mathcal{L}_{\mathrm{reg}} + w_{\mathrm{kd}} \mathcal{L}_{\mathrm{kd}} + w_{\mathrm{kdf}} \mathcal{L}_{\mathrm{kdf}}, \tag{8}$$

where $w_1, w_{\text{reg}}, w_{\text{kd}}, w_{\text{kdf}}$ are the weights to balance different loss terms, respectively; $\mathcal{L}_1$ is the pixel-wise L1 loss between the output image and the ground truth; $\mathcal{L}_{\text{reg}}$ is a bit regularization loss ($\mathcal{L}_{\text{wb}}$ in our case); $\mathcal{L}_{\text{kd}}$ is the knowledge distillation loss on the last output feature using 8-bit quantized model as the teacher; $\mathcal{L}_{\text{kdf}}$ is the knowledge distillation loss on output feature of each layer using the same 8-bit teacher. As for the knowledge distillation, a teacher network has the same architecture backbone (*i.e.*, no bit selection module included) as the student SR network. A teacher network is pre-trained with uniform 8-bit weights with the activations quantized via PAMS [31].

4 Experiments

The proposed quantization framework CADyQ is evaluated on various SR networks to validate its effectiveness and flexibility. We first describe our experimental settings (Sec. 4.1) and evaluate our method on various SR networks (Sec. 4.2). We then present detailed ablation experiments to analyze each main attribute of our framework (Sec. 4.4): namely, layer-wise/patch-wise quantization, quantization sensitivity estimation, and the proposed weighted bit loss.

4.1 Implementation Details

Models. The proposed framework is applied directly to existing SR networks, including representative SR networks (EDSR-baseline [34] and SRResNet [29]) and recent efficient models (IDN [21] and CARN [2]), thereby naming the CADyQ-quantized models as EDSR-baseline-CADyQ, SRResNet-CADyQ, IDN-CADyQ, and CARN-CADyQ, respectively. Following the settings from previous works on SR quantization [31,38,46], our framework quantizes weights and feature maps in the layers of the high-level feature extraction module where most of the costly operations are concentrated in. In this work, we set the quantization bit-width candidates as $\{4, 6, 8\}$ and employ linear symmetric quantization function [31] with a learnable scale parameter for each quantization function candidate. Furthermore, we uniformly apply 8-bit linear quantization for weights. Additional experiments that demonstrate the applicability of CADyQ are provided in supplementary document Sec. A.

Training Details. Training and validation are done with DIV2K [1] dataset. For training stability, we follow [25,52] in initializing the SR network parameters with pre-trained 8-bit network weights and in controlling the bit selector to progressively decrease the bit-width. For a progressive reduction in bit-width, the weight of the proposed bit regularization loss (w_{reg}) is initially 10^{-4} and gradually increased throughout training (10^{-6} per $1K$ iteration). The weights for the loss terms w_1, w_{kd}, and w_{kdf} is respectively 1.0, 1000.0, and 100.0. Analysis on w_{reg} and other training settings are specified in Sec. C of the supplementary document.

Table 1: Quantitative comparisons on various SR networks: IDN [21], EDSR-baseline [34], SRResNet [29], and CARN [2] of scale 4. The average feature quantization rate of the feature extraction stage (FQR), PSNR, and SSIM are reported. The results demonstrate the efficiency of the proposed method that manages to reduce FQR while maintaining or improving PSNR/SSIM

Model	Urban100			Test2K			Test4K		
	$FQR_\downarrow$	$PSNR_\uparrow$	$SSIM_\uparrow$	$FQR_\downarrow$	$PSNR_\uparrow$	$SSIM_\uparrow$	$FQR_\downarrow$	$PSNR_\uparrow$	$SSIM_\uparrow$
IDN [21]	32.00	25.42	0.763	32.00	27.48	0.774	32.00	28.54	0.806
IDN-PAMS [31]	8.00	25.56	0.768	8.00	27.53	0.775	8.00	28.59	0.807
IDN-DAQ [17]	4.00	24.46	0.718	4.00	26.98	0.750	4.00	27.94	0.782
IDN-CADyQ (Ours)	5.78	25.65	0.771	5.16	27.54	0.776	5.03	28.61	0.808
EDSR-baseline [34]	32.00	26.04	0.784	32.00	27.71	0.782	32.00	28.80	0.814
EDSR-baseline-PAMS [31]	8.00	25.94	0.781	8.00	27.67	0.781	8.00	28.77	0.813
EDSR-baseline-DAQ [17]	4.00	25.73	0.772	4.00	27.60	0.777	4.00	28.67	0.809
EDSR-baseline-CADyQ (Ours)	6.09	25.94	0.782	5.52	27.67	0.781	5.37	28.77	0.813
SRResNet [29]	32.00	25.74	0.773	32.00	27.60	0.778	32.00	28.68	0.810
SRResNet-PAMS [31]	8.00	25.85	0.776	8.00	27.63	0.779	8.00	28.72	0.812
SRResNet-DAQ [17]	4.00	25.70	0.772	4.00	27.59	0.778	4.00	28.67	0.810
SRResNet-CADyQ (Ours)	5.73	25.92	0.781	5.14	27.64	0.781	5.02	28.72	0.812
CARN [2]	32.00	26.07	0.784	32.00	27.69	0.782	32.00	28.79	0.814
CARN-PAMS [31]	8.00	25.80	0.776	8.00	27.60	0.778	8.00	28.68	0.811
CARN-DAQ [17]	4.00	25.48	0.764	4.00	27.30	0.771	4.00	28.24	0.802
CARN-CADyQ (Ours)	5.32	25.94	0.780	4.65	27.65	0.780	4.54	28.73	0.812

Evaluation Details. We evaluate our framework on the standard benchmark (Urban100 [19]) and on more computationally demanding images of large size (e.g., 2K, 4K) in Test2K and Test4K datasets [28] which are generated via bicubic downsampling from DIV8K [14] dataset (index 1201-1400). For testing, an input test image is cropped into patches of size 96×96 with six overlapping boundary pixels. Each patch is super-resolved with our framework and then combined to produce the whole HR image. There exists a trade-off between the patch size and the overall efficiency, which is discussed in Sec. 4.5 and supplementary document Sec. B. We report peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM [44]) to evaluate the SR performance, along with the average feature quantization rate (FQR) for the evaluation of efficiency. Furthermore, our ablation study is conducted consistently with CARN backbone models on Urban100 dataset.

4.2 Quantitative Results

To evaluate the effectiveness and efficiency of the proposed mechanism, we compare the results with PAMS [31] and DAQ [17] using the official code, which is similar to CADyQ in that quantization is directly applied to existing SR networks without redesigning the architecture. Specifically, we compare with PAMS (8-bit) since lower-bit quantization by PAMS results in performance degradation, and DAQ (w4a4qq4) with 4-bit for weight and feature quantization. As

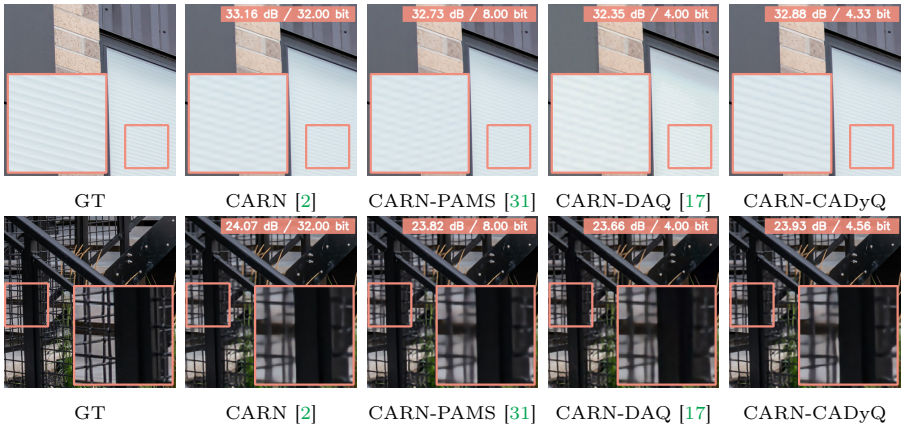


Fig. 4: Qualitative results on 'img1215' of Test4K. Quantitative measures of PSNR and average bit-width of the patch are also reported (PSNR / Average bit-width). More results are provided in Section E of the supplementary document

shown in Table 1, DAQ reduces the computational resources but at the cost of performance degradation, which is especially severe (over -0.5dB) for IDN and CARN baseline. Also, compared with PAMS, CADyQ demonstrates a lower average precision without the performance drop, striking a better balance between computational cost and performance.

4.3 Qualitative Results

Fig. 4 provides qualitative results and comparisons with the output images from CARN-based models [2]. CARN-CADyQ (ours) produces a visually clean output image, while CARN-PAMS and sometimes even original unquantized CARN suffer from a checkerboard artifact or blurred lines, even though CARN-CADyQ uses less computational resources. Moreover, CARN-DAQ, despite the low computational resources, produces various artifacts and color distortion. Also, the map of average bit-width used by our framework for each local patch in the image is visualized in Fig. 5 (a). The visualized bit map demonstrates that our framework allocates more computational resources to patches with complex structures (e.g., buildings) and less to patches with simple structures (e.g., sky). Furthermore, CADyQ is shown to dynamically allocate distinct bit-widths across different network layers, as visualized in Fig. 5 (b). The qualitative results stress the effectiveness and importance of the patch-and-layer-wise bit allocation.

4.4 Ablation Study

Effect of Layer-Wise and Patch-Wise Quantization. To verify the importance of layer-wise and patch-wise quantization in conjunction, we compare our

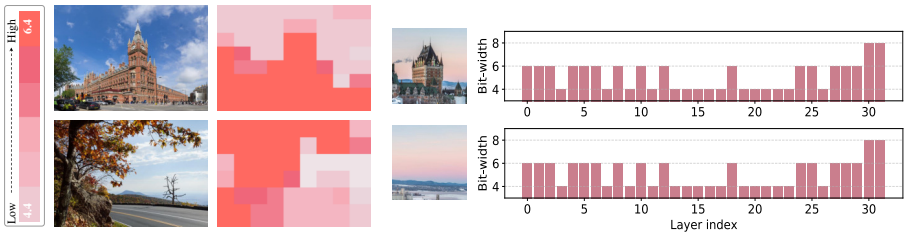


Fig. 5: Visualizations of dynamic bit-width allocation by CADyQ across patches and layers. On average, CADyQ assigns higher bit-width to **(a)** complex patches and to **(b)** important layers. Results are obtained with EDSR-CADyQ from (a) 'img1215' and 'img1222' from Test4K and (b) two patches of 'img1400' from Test2K

Table 2: Ablation study on layer-wise and patch-wise quantization

	Layer-wise	Patch-wise	FQR$_\downarrow$	PSNR$_\uparrow$	SSIM$_\uparrow$
(2a)	✗	✗	8.00	25.80	0.776
(2b)	✗	✓	6.15	25.89	0.778
(2c)	✓	✗	7.02	25.92	0.780
CADyQ	✓	✓	5.32	25.94	0.780

overall scheme CADyQ with its separate modules individually: patch-wise quantization and layer-wise quantization, as reported in Table 2. Quantization with patch-wise dynamic bit-width but fixed throughout the network (model **(2b)**) results in a performance drop. A layer-wise different bit but fixed across different patches and images (model **(2c)**) preserves the restoration accuracy but with a small improvement in efficiency (average 7.02 bit). By contrast, layer-wise and patch-wise quantization in conjunction effectively enhances the quality of the super-resolved image and reduces the average bit-width by a large amount. The results validate our claim that dynamically determining the bit-width both per layer and patch is important, corroborating our observations from Fig. 2.

Quantization Sensitivity Estimation. In this ablation study, we validate our choice of measures for quantization sensitivity, which a bit selector uses to decide the bit-width for each patch and layer, as shown in Table 3. We compare alternative measures that could estimate the quantization sensitivity and have similar computational overheads to our choice: patch gradient and channel-wise standard deviation. Although utilizing the range (the gap between max and min value) of the patch pixel values and the layer-wise feature (model **(3a)**) requires fewer computations, it induces a severe performance degradation. Also, the standard deviation of a patch and the layer-wise standard deviation of the feature (model **(3b)**) suffers from performance degradation. Notably, using the layer-wise standard deviation (model **(3c)**) results in lower performance (PSNR

Table 3: Ablation study on quantization sensitivity measures

	Patch	Layer	FQR$_\downarrow$	PSNR$_\uparrow$	SSIM$_\uparrow$
(3a)	max-min	max-min	4.51	25.16	0.752
(3b)	std	layer std	5.95	25.62	0.769
(3c)	gradient	layer std	6.53	25.80	0.775
CADyQ	gradient	channel std	5.32	25.94	0.780

Table 4: Ablation study on losses: bit loss and knowledge distillation loss

	Loss				FQR$_\downarrow$	PSNR$_\uparrow$	SSIM$_\uparrow$
	$\mathcal{L}_1$	$\mathcal{L}_{\text{reg}}$	$\mathcal{L}_{\text{kd}}$	$\mathcal{L}_{\text{kdf}}$			
(4a)	✓	✗	✓	✓	6.51	25.70	0.772
(4b)	✓	$\mathcal{L}_{\text{b}}$	✓	✓	5.75	25.68	0.772
(4c)	✓	$\mathcal{L}_{\text{wb}}$	✗	✓	4.48	25.38	0.761
(4d)	✓	$\mathcal{L}_{\text{wb}}$	✓	✗	5.25	25.51	0.766
CADyQ	✓	$\mathcal{L}_{\text{wb}}$	✓	✓	5.32	25.94	0.780

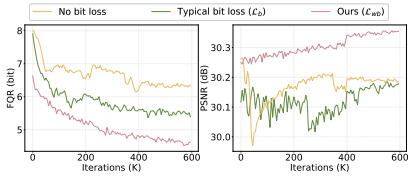


Fig. 6: Learning curves with different losses for FQR (left) and PSNR (right)

or SSIM) and higher average precision (FQR), compared with using channel-wise standard deviation (CADyQ). We provide further justifications for using standard deviations of each channel in the supplementary document Sec. B. In summary, the ablation study implies that the patch gradient and channel-wise standard deviation of the feature contain important information that gives a better estimate of the quantization sensitivity, thereby helping to find a better trade-off between performance and computational resources.

Ablation on Losses. We analyze the effect of the proposed weighted bit loss $\mathcal{L}_{\text{wb}}$ (Eq. (7)) by removing it (model **(4a)**) or replacing it with the conventional bit loss $\mathcal{L}_{\text{b}}$, formulated in Eq. (6), (model **(4b)**) in the overall objective function from Eq. (8), as displayed in Table 4 and Fig. 6. The figure shows the curves of the average bit-widths and PSNR of the validation dataset during training. Without the bit loss, the framework fails to reduce the computational resources effectively. Also, replacing our bit loss with the conventional bit loss reduces the average bit-width but with ~ 0.2 dB PSNR drop. On the other hand, our proposed bit loss substantially reduces the computational resources without a PSNR drop. Then we evaluate the effect of knowledge distillation loss by removing the distillation loss on the output feature (model **(4c)**) and intermediate features (model **(4d)**), respectively. The results in the table suggest that both types of knowledge distillation play key roles in maintaining the restoration performance.

4.5 Complexity Analysis

Our framework can process either the full input test image at once or process, in parallel, the smaller patches, which are combined to construct the full image. For both scenarios, we analyze the complexity of our framework w.r.t. the number

Table 5: Complexity analysis of image-wise and patch-wise inference measured w.r.t. BitOPs of the feature extraction stage on CARN backbone models

Patch Size	Full Image			96×96		
Model	Baseline	PAMS	**CADyQ**	Baseline	PAMS	**CADyQ**
PSNR$_\uparrow$(dB)	26.07	25.80	**25.95**	26.07	25.80	**25.94**
BitOPs$_\downarrow$(G)	76.44	4.78	**3.24**	77.87	4.87	**3.23**

Table 6: Average GPU inference latency. GPU latency for each model is measured with EDSR backbone models on Test4K images

Model	Baseline	PAMS	**CADyQ**
GPU Inference Latency (ms)	535.5	240.0	**206.5**

of operations weighted by the bit-widths of the operands (BitOPs) for generating a 720p (1820 × 720) image in Table 5. We use BitOPs as measurements for the computational complexity to better reflect the reduction in bit-width. Our framework is shown to be more effective on the patch-wise inference, as local regions with complex structures and those with simple structures are processed with different computational resources. Despite additional computational overhead from overlapping area between neighboring patches in the patch-wise inference, our framework achieves ∼95.8% reduction in BitOPs compared with the baseline and ∼32.2% reduction in BitOPs compared with 8-bit CARN-PAMS at image-wise inference. GPU latency is measured on NVIDIA Tesla T4 GPU with Tensor Cores supporting 4/8-bit acceleration. As hardware-acceleration for 6-bit is not supported on T4, we cast 6-bit assignments as 8-bit. On average, inference latency of Test4K images is improved to 206.5ms for CADyQ, compared with 240.0ms for 8-bit quantization (PAMS [31]), and 535.5ms for 32-bit. Computational complexity of other backbone models and detailed analysis of overheads can be found in Sec. D of the supplementary document.

5 Conclusion

In this work, we study and exploit the relationship between the local image contents (*e.g.*, local patches and their features at each layer) and the super-resolution performance degradation from quantization. We thereby propose a patch-and-layer-wise bit allocation method for dynamic quantization. Experimental results demonstrate that the proposed quantization framework, CADyQ manages to reduce the computational complexity with respect to BitOPs and inference latency with negligible performance drop.

Acknowledgment

This work was supported in part by the IITP grant funded by the Korea government (MSIT) [No. 2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University), No. 2021-0-02068, Artificial Intelligence Innovation Hub, and No.2022-0-00156], and in part by the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2022.

References

1. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: CVPR Workshops (2017) 9
2. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: ECCV (2018) 2, 9, 10, 11
3. Ayazoglu, M.: Extremely lightweight quantization robust real-time single-image super resolution for mobile devices. In: CVPR Workshops (2021) 4
4. Bengio, Y., Léonard, N., Courville, A.: Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432 (2013) 7
5. Cai, Z., He, X., Sun, J., Vasconcelos, N.: Deep learning with low precision by half-wave gaussian quantization. In: CVPR (2017) 4, 5
6. Cai, Z., Vasconcelos, N.: Rethinking differentiable search for mixed-precision neural networks. In: CVPR (2020) 4, 8
7. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.J., Srinivasan, V., Gopalakrishnan, K.: Pact: Parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085 (2018) 2, 4, 5
8. Chu, X., Zhang, B., Ma, H., Xu, R., Li, Q.: Fast, accurate and lightweight super-resolution with neural architecture search. In: ICPR (2021) 4
9. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV (2014) 4
10. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE TPAMI **38**(2), 295–307 (2015) 1
11. Dong, Z., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Hawq: Hessian aware quantization of neural networks with mixed-precision. In: ICCV (2019) 4
12. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization. arXiv preprint arXiv:1902.08153 (2019) 4
13. Fattal, R.: Image upsampling via imposed edge statistics. TOG (2007) 3, 7
14. Gu, S., Lugmayr, A., Danelljan, M., Fritsche, M., Lamour, J., Timofte, R.: Div8k: Diverse 8k resolution image dataset. In: ICCV Workshops (2019) 10
15. Habi, H.V., Jennings, R.H., Netzer, A.: Hmq: Hardware friendly mixed precision quantization block for cnns. In: ECCV (2020) 4
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 8
17. Hong, C., Kim, H., Baik, S., Oh, J., Lee, K.M.: Daq: Channel-wise distribution-aware quantization for deep image super-resolution networks. In: WACV (2022) 5, 10, 11
18. Hou, L., Kwok, J.T.: Loss-aware weight quantization of deep networks. In: ICLR (2018) 2
19. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR (2015) 10
20. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: ACMMM (2019) 4
21. Hui, Z., Wang, X., Gao, X.: Fast and accurate single image super-resolution via information distillation network. In: CVPR (2018) 4, 9, 10
22. Ignatov, A., Timofte, R., Denna, M., Younes, A.: Real-time quantized image super-resolution on mobile npus, mobile ai 2021 challenge: Report. In: CVPR Workshops (2021) 2, 4

23. Jiang, X., Wang, N., Xin, J., Li, K., Yang, X., Gao, X.: Training binary neural network without batch normalization for image super-resolution. In: AAAI (2021) 4
24. Jin, Q., Yang, L., Liao, Z.: Adabits: Neural network quantization with adaptive bit-widths. In: CVPR (2020) 4
25. Jung, S., Son, C., Lee, S., Son, J., Han, J.J., Kwak, Y., Hwang, S.J., Choi, C.: Learning to quantize deep networks by optimizing quantization intervals with task loss. In: CVPR (2019) 4, 5, 9
26. Kim, H., Hong, S., Han, B., Myeong, H., Lee, K.M.: Fine-grained neural architecture search. arXiv preprint arXiv:1911.07478 (2019) 4
27. Kim, J., Lee, J., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016) 1
28. Kong, X., Zhao, H., Qiao, Y., Dong, C.: Classsr: A general framework to accelerate super-resolution networks by data characteristic. In: CVPR (2021) 4, 10
29. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR (2017) 4, 9, 10
30. Lee, J., Kim, D., Ham, B.: Network quantization with element-wise gradient scaling. In: CVPR (2021) 4
31. Li, H., Yan, C., Lin, S., Zheng, X., Zhang, B., Yang, F., Ji, R.: Pams: Quantized super-resolution via parameterized max scale. In: ECCV (2020) 2, 4, 5, 6, 7, 9, 10, 11, 14
32. Li, Y., Gu, S., Zhang, K., Gool, L.V., Timofte, R.: Dhp: Differentiable meta pruning via hypernetworks. In: ECCV (2020) 4
33. Li, Y., Li, W., Danelljan, M., Zhang, K., Gu, S., Van Gool, L., Timofte, R.: The heterogeneity hypothesis: Finding layer-wise differentiated network architectures. In: CVPR (2021) 4
34. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPR Workshops (2017) 1, 4, 9, 10
35. Liu, J., Wang, Q., Zhang, D., Shen, L.: Super-resolution model quantized in multi-precision. Electronics **10**(17), 2176 (2021) 5
36. Liu, M., Zhang, Z., Hou, L., Zuo, W., Zhang, L.: Deep adaptive inference networks for single image super-resolution. In: ECCV (2020) 4
37. Lou, Q., Guo, F., Liu, L., Kim, M., Jiang, L.: AutoQ: Automated kernel-wise neural network quantization. In: ICLR (2020) 4
38. Ma, Y., Xiong, H., Hu, Z., Ma, L.: Efficient super resolution using binarized neural network. In: CVPR Workshops (2019) 2, 4, 9
39. Oh, J., Kim, H., Nah, S., Hong, C., Choi, J., Lee, K.M.: Attentive fine-grained structured sparsity for image restoration. In: CVPR (2022) 4
40. Son, S., Lee, K.M.: Srwarp: Generalized image super-resolution under arbitrary transformation. In: CVPR (2021) 1
41. Wang, H., Chen, P., Zhuang, B., Shen, C.: Fully quantized image super-resolution networks. In: ACMMM (2021) 5, 6, 7
42. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: Haq: Hardware-aware automated quantization with mixed precision. In: CVPR (2019) 4
43. Wang, L., Dong, X., Wang, Y., Ying, X., Lin, Z., An, W., Guo, Y.: Exploring sparsity in image super-resolution for efficient inference. In: CVPR (2021) 4
44. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004) 10

45. Xie, W., Song, D., Xu, C., Xu, C., Zhang, H., Wang, Y.: Learning frequency-aware dynamic network for efficient super-resolution. In: ICCV (2021) 4
46. Xin, J., Wang, N., Jiang, X., Li, J., Huang, H., Gao, X.: Binarized neural network for single image super resolution. In: ECCV (2020) 4, 9
47. Yang, L., Jin, Q.: Fracbits: Mixed precision quantization via fractional bit-widths. In: AAAI (2021) 4, 8
48. Yu, K., Wang, X., Dong, C., Tang, X., Loy, C.C.: Path-restore: Learning network path selection for image restoration. IEEE TPAMI (2021) 4
49. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: ECCV (2018) 1, 4
50. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: CVPR (2018) 1
51. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016) 2, 4
52. Zhuang, B., Shen, C., Tan, M., Liu, L., Reid, I.: Towards effective low-bitwidth convolutional neural networks. In: CVPR (2018) 4, 9