SCALES: Boost Binary Neural Network for Image Super-Resolution with Efficient Scalings

Renjie Wei[12], Zechun Liu[5], Yuchen Fan[5], Runsheng Wang[234], Ru Huang[234], and Meng Li[123†]
[1]*Institute for Artificial Intelligence &* [2]*School of Integrated Circuits, Peking University, Beijing, China*
[3]*Beijing Advanced Innovation Center for Integrated Circuits, Beijing, China*
[4]*Institute of Electronic Design Automation, Peking University, Wuxi, China* [5]*Meta Inc Menlo Park CA, USA*

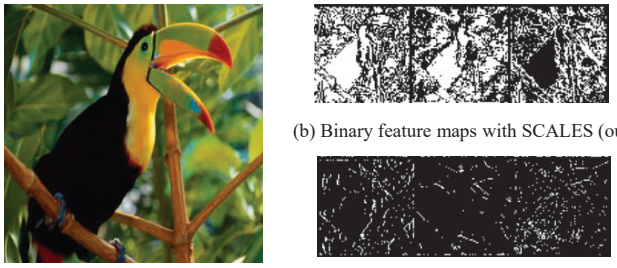
***Abstract*—Deep neural networks for image super-resolution (SR) have demonstrated superior performance. However, the large memory and computation consumption hinders their deployment on resource-constrained devices. Binary neural networks (BNNs), which quantize the floating point weights and activations to 1-bit can significantly reduce the cost. Although BNNs for image classification have made great progress these days, existing BNNs for SR still suffer from a large performance gap between the FP SR networks. To this end, we observe the activation distribution in SR networks and find much larger pixel-to-pixel, channel-to-channel, layer-to-layer, and image-to-image variation in the activation distribution than image classification networks. However, existing BNNs for SR fail to capture these variations that contain rich information for image reconstruction, leading to inferior performance. To address this problem, we propose SCALES, a binarization method for SR networks that consists of the layer-wise scaling factor, the spatial re-scaling method, and the channel-wise re-scaling method, capturing the layer-wise, pixel-wise, and channel-wise variations efficiently in an input-dependent manner. We evaluate our method across different network architectures and datasets. For CNN-based SR networks, our binarization method SCALES outperforms the prior art method by 0.2dB with fewer parameters and operations. With SCALES, we achieve the first accurate binary Transformer-based SR network, improving PSNR by more than 1dB compared to the baseline method.**

***Index Terms*—Binary neural network, image super-resolution, layer-wise scaling factor, spatial re-scaling, channel-wise re-scaling**

I. Introduction

Image super-resolution (SR) is a fundamental task in computer vision. It aims to reconstruct high-resolution (HR) images, which have more details and high-frequency information, from low-resolution (LR) images. In recent years, deep neural networks (DNNs) have achieved great quality in image SR including convolution neural network (CNN)-based [1]–[4] and Transformer-based [5]–[7] methods. However extensive parameters and computation demands of these SR networks hinder their deployment on resource-constrained devices.

Network quantization is always a turn-to solution to reduce memory and computation costs. Among these, binary neural networks (BNN) quantizing the full-precision (FP) weights and activations to 1-bit can achieve $32\times$ memory savings and $58\times$ speed up on CPUs [8], which is quite effective. However, there is still a lack of research on BNN for SR, compared with BNN for image classification. On the one hand, for CNN-based SR methods, BNN [9]–[11] still suffers from large performance degradation compared with their FP counterpart. On the other hand, for Transformer-based SR methods, there is no research on BNN to the best of our knowledge.



(a) HR image (b) Binary feature maps with SCALES (ours) (c) Binary feature maps with E2FIF (the prior art)

Fig. 1: The binary feature maps with our method SCALES and the prior art method E2FIF.

To this end, we study the BNN for SR comprehensively for both CNN-based and Transformer-based SR networks. We discover that the activation distributions in FP SR networks including CNN and Transformer exhibit much larger pixel-to-pixel, channel-to-channel, layer-to-layer, and image-to-image variations compared to the image classification networks. Exiting BNNs for SR can not capture these variations that contain detailed information for image SR simply using the binarization methods for classification networks. To avoid such important variations getting lost during binarization, we propose our method SCALES, which consists of the layer-wise scaling factor, the spatial re-scaling, and channel-wise re-scaling method, to capture the layer-wise, pixel-wise, and channel-wise variations respectively in an input-dependent manner. With our method, we can preserve more textures and details for image SR compared to the prior art method in Fig. 1. Extensive experiments demonstrate the effectiveness of our method. For example, for CNN-based networks, SCALES outperforms the prior art method by 0.22dB and 0.19dB on Urban100 dataset with less number of parameters and operations. For Transformer-based networks, SCALES improves PSNR by more than 1dB compared to the baseline, leading to the first accurate binary Transformer-based SR network.

Overall, our contributions can be summarized as follows:

- We observe the activation distribution in the CNN-

This work was supported in part by National Natural Science Foundation of China under Grant 62495102 and Grant 92464104, in part by Beijing Municipal Science and Technology Program under Grant Z241100004224015, and in part by 111 Project under Grant B18001.

†Corresponding author.

978-3-9826741-0-0/DATE25/© 2025 EDAA

based and Transformer-based SR networks and discover large pixel-to-pixel, channel-to-channel, layer-to-layer, and image-to-image variations, which are important for high-performance image SR.
- To capture the variations, we propose a binarization method for SR networks, dubbed SCALES, which is composed of the layer-wise scaling factor, the spatial re-scaling method, and the channel-wise re-scaling method.
- We evaluate SCALES across different SR network architectures on different benchmark datasets. For CNN-based SR networks, SCALES outperforms the prior art method by 0.2dB with fewer parameters and operations. With SCALES, we also achieve the first accurate binary Transformer-based SR network, improving PSNR by more than 1dB compared to the baseline method.

II. Background

A. DNNs for Image SR

DNNs have been widely used in image SR for their satisfying performance. SRCNN [12] first uses three convolution layers to reconstruct the HR image in an end-to-end way. VDSR [13] increases the network depth to 20 convolution layers and introduces global residual learning. SRResNet [1] introduces residual blocks as the basic block. EDSR [2] removes batch normalization layers (BN) in the basic block and uses a deeper and wider model, which has become the standard architecture for CNN-based SR networks. After that, dense connect [3], channel attention module [14], and non-local attention mechanism [15] are introduced in SR networks for better image quality. In recent years, Transformer-based SR networks are proposed with better performance. IPT [5] leverages the Transformer encoder-decoder structure and pre-training on large-scale dataset. SwinIR [6] is based on Swin Transformer and performs better than IPT. HAT [7] combines channel attention, window-based self-attention, and cross-attention schemes and reaches better image SR performance.

The architecture of typical DNNs for SR is shown in Fig. 2. It consists of three modules. The head module extract shallow features from the input LR image. The body module utilizes multiple basic blocks to perform deep feature extraction and the deep feature is fused with shallow feature through global residual connection. CNN-based and Transformer-based networks use different basic blocks as shown in Fig. 2. The former incorporates convolution layers and ReLU. The latter incorporates transformer layers including layernorm, multi-head self-atteion (MSA), multi-layer perceptron (MLP) and the additional convolution layer. The tail module reconstruct the high-resolution output by convolution and pixel shuffling.

B. BNNs for SR

BNN, which quantizes both activation and weight to $\{-1, 1\}$ is first proposed by XNOR-Net [8] for the image classification task. Afterward, a lot of works [16]–[22] are proposed to reduce the classification accuracy gap between BNNs and their FP counterparts. However, research on BNNs for SR is relatively scarce. Table I lists some representative works. It is worth noting that they are all designed for CNN-based networks. [23] first introduces binarization to SR networks and reduces the model size of FP SRResNet. However, they only binarize weights and leave activations at FP, which impedes the bit-wise operation and requires expensive FP accumulations. BAM [9] binarizes both weights and activations using a bit-accumulation mechanism to approximate the FP convolution. They binarize weights and activations in each layer based on the accumulation of previous layers, which introduces extra FP accumulation during inference. BTM [10] finds that BN in BNNs introduces a lot of FP calculations. They design a binary training mechanism to normalize input LR images and build a BNN without BN, named IBTM. LMB [24] calculates the threshold for each pixel by averaging its neighborhood pixel values, which increases computation significantly for calculating per-pixel threshold. DAQ [25] proposes a per-channel activation quantization method to adapt the diverse channel-wise distributions. However, it introduces large FP computations for calculating the mean and standard deviation of each channel of activations. E2FIF [11] proposes an end-to-end full-precision information flow to improve the performance of BNN. However, these methods still exhibit a performance gap compared to their FP counterparts because they fail to capture the large activation variations. Moreover, how to binarize Transformer-based SR networks remains an unresolved issue.

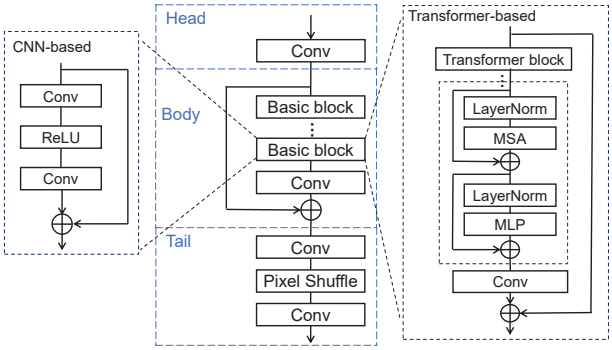


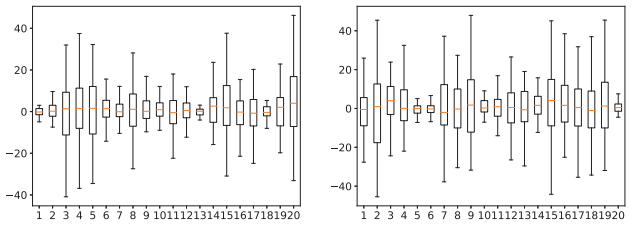
Fig. 2: The typical architecture of CNN-based and Transformer-based SR networks and the detailed structure of basic blocks.

TABLE I: Comparison between existing BNNs for SR and our work on the spatial, channel-wise, layer-wise, image-wise adaptability, and the hardware cost.

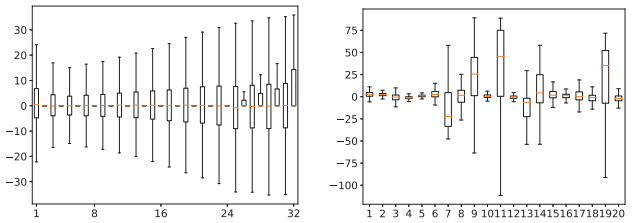
Method	Spa. Adpt.	Chl. Adpt.	Layer Adpt.	Img. Adpt.	HW cost
[23]	✗	✗	✗	✗	FP Accum.
BAM [9]	✔	✗	✗	✗	Extra FP Accum.
BTM [10]	✗	✗	✗	✔	Low
LMB [24]	✔	✗	✗	✔	FP Accum.
DAQ [25]	✗	✔	✗	✔	FP Mul. and Accum.
E2FIF [11]	✗	✗	✗	✗	Low
SCALES (ours)	✔	✔	✔	✔	Low

III. Motivation

Existing BNNs focus on binarizing the weights and the input activations of the basic blocks in the body module as shown in Fig. 2, which account for most of the parameters and computations of the entire model. However, they ignore the large variations in activation distribution and have inferior SR

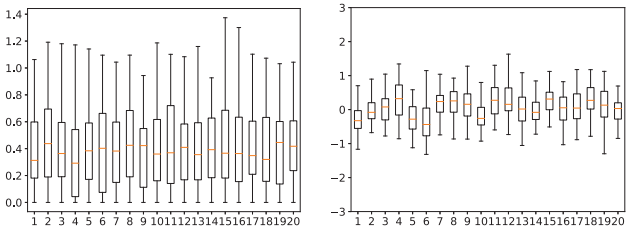


(a) Distribution across pixels (img1) (b) Distribution across pixels (img2)



(c) Distribution across layers (d) Distribution across channels

Fig. 3: Activation distribution in EDSR [2].



(a) Distribution in ResNet18 (b) Distribution in SwinViT

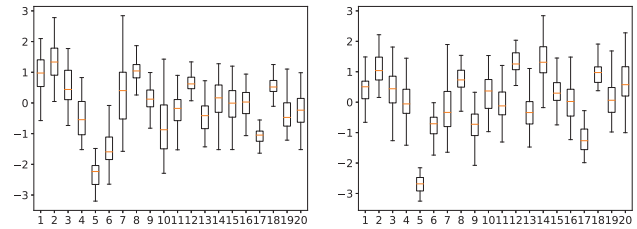
Fig. 4: Activation distribution in CNN-based and Transformer-based classification networks ResNet18 [26] and SwinViT [27]. The distributions across pixels, channels, layers, and images are similar, thus we only show distributions across pixels here.

performance. In this section, we showcase that the activation distributions in FP SR networks exhibit much larger *pixel-to-pixel, channel-to-channel, layer-to-layer, and image-to-image variations* than those in the image classification networks.

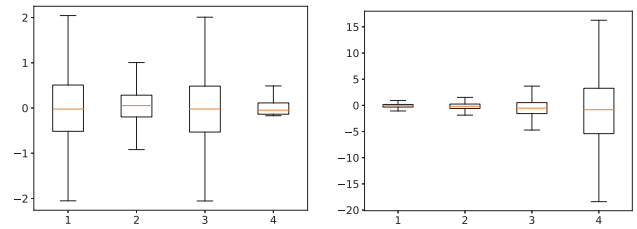
A. Variations in CNN-based SR Network

As shown in Fig. 3a, we random sample 20 pixels from a feature map in a CNN-based SR network EDSR, where each pixel contains C (the number of channels) elements. We observe large *pixel-to-pixel variation*, compared to ResNet18 in Fig. 4a. The same holds true for *channel-to-channel variation*. The main reason is that modern SR networks removes BN for better SR performance, leading to large activation variations. According to [28], the difference in activation magnitudes indicates different scaling factors are needed. However, per-pixel or per-channel quantization for activation is infeasible because they will introduce large computation overhead [29] while existing per-tensor binarization schemes cannot capture the variation of activation distribution.

For different layers, activations in EDSR also exhibit large *layer-to-layer variation* in Fig. 3c. Moreover, we find that the



(a) Distribution across pixels (img1) (b) Distribution across pixels (img2)



(c) Distribution across layers (linear) (d) Distribution across layers (conv)

Fig. 5: Activation distribution in SwinIR [6].

TABLE II: Activation variance comparison.

	EDSR	ResNet	SwinIR	SwinViT
chl-to-chl	439.17	0.10	0.11	0.10
pixel-to-pixel	622.25	0.34	0.87	0.12
layer-to-layer	3494.38	0.92	162.70	3.46
image-to-image	599.39	0.32	0.84	0.13

activations in the even layers are small, whereas the activations in the odd layers exhibit large magnitudes. This is because for the basic block in Fig. 2, the shortcut maintains the original information of the input LR image, while the inner branch intends to re-construct the small difference between the LR and HR image. Thus, the input of the first conv layer has large magnitude, while the input of the second conv layer has small magnitude, which implies that different binarization schemes should be employed for different layers. Comparing Fig. 3a and 3b, we can also find the large *image-to-image variation*, which motivates us to quantize the SR network in an input-dependent manner.

B. Variations in Transformer-based SR Network

Large *pixel-to-pixel, layer-to-layer* and *image-to-image variations* also exist in Transformer-based SR networks as shown in Fig. 5. For the layer-to-layer variation, we plot the input activations of the four linear layers in the Transformer block in Fig. 5c and the last conv layer of each basic block in Fig. 5d. It is worth noting that channel-to-channel variation does not exist in SwinIR, since LayerNorm (LN) normalizes each token across the channel dimension. Thus, for the Transformer-based SR network, we should apply different quantization schemes to capture the pixel-wise, layer-wise, and image-wise variations. For quantitative comparison, we calculate the variance of image SR and classification networks in Table II, which is consistent with our observations above.

IV. METHOD

In this section, we introduce our proposed method SCALES, which mainly consists of three components, including the layer-wise scaling factor, the spatial re-scaling module, and the channel-wise re-scaling module capturing the layer-to-layer, pixel-to-pixel, and channel-to-channel variations efficiently in an input-dependent manner.

A. Layer-wise Scaling Factor

Currently, existing BNNs for SR either use complicated binarization functions [24], [25], [30] which have expensive computation costs, or use the sign function for activation binarization [11], i.e., $\hat{x} = \text{sign}(x)$ which can not capture the variations in SR networks. To this end, we propose using the layer-wise scaling factor α to capture the layer-to-layer variation. Each convolution or linear layer has a scaling factor, which is learned to have different magnitudes for different layers. We further introduce the channel-wise threshold β that is also learnable inspired by ReActNet [19] to capture the channel-wise shifting in Fig. 3d. Thus, our activation binarization function becomes:

$$\hat{x} = \alpha \,\text{sign}(\frac{x-\beta}{\alpha}), \tag{1}$$

where β is the channel-wise learnable threshold, and α is the layer-wise scaling factor. Both can be optimized end-to-end with the gradient-based method with the help of the straight through estimator (STE) [17]. The gradient w.r.t. α can be calculated as:

$$\frac{\partial \hat{x}}{\partial \alpha} = \begin{cases} -1, & \text{if } x \leq \beta - \alpha \\ -2\left(\frac{x-\beta}{\alpha}\right)^2 - 2\frac{x-\beta}{\alpha} - 1, & \text{if } \beta - \alpha < x \leq \beta \\ 2\left(\frac{x-\beta}{\alpha}\right)^2 - 2\frac{x-\beta}{\alpha} + 1, & \text{if } \beta < x \leq \beta + \alpha \\ 1, & \text{if } x > \beta + \alpha \end{cases} \tag{2}$$

while the gradient w.r.t. β can be computed as:

$$\frac{\partial \hat{x}}{\partial \beta} = \begin{cases} -2 - 2\frac{x-\beta}{\alpha}, & \text{if } \quad \beta - \alpha < x \leqslant \beta \\ -2 + 2\frac{x-\beta}{\alpha}, & \text{if } \beta < x \leqslant \beta + \alpha \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

For weights, we binarize them in a per-channel way as usual: $\hat{w} = \frac{\|w\|_{l1}}{n}\,\text{sign}(w)$, where n denotes the number of weights. The scaling factor is the absolute mean value for each output channel.

B. Spatial Re-scaling

To capture the pixel-to-pixel variation in SR networks, we propose the spatial re-scaling method for both CNN-based networks in Fig. 6a and Transformer-based networks in Fig. 6b.

We use the FP activation before binarization as the input and predict the spatial scaling factors to re-scale the output of binary convolution or binary linear layer. The spatial scaling factors are predicted through the right-hand side branch, i.e., the FP convolution layer with 1×1 kernel and sigmoid layer in Fig. 6a and the FP linear layer and sigmoid layer in Fig. 6b. Although they are in FP, they only introduce little parameters compared to the original parameters of binary conv or linear layer. It is also worth noting that during inference the spatial scaling factor is not fixed but inferred from data. Thus the spatial re-scaling module can capture spatial information in an input-dependent manner. As a result, the pixel-to-pixel and image-to-image variations are well captured. Our spatial re-scaling method is formulated as:

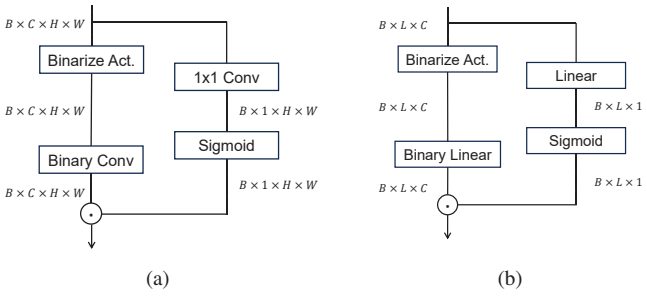


Fig. 6: The proposed spatial re-scaling method for (a) CNN-based and (b) Transformer-based SR network.

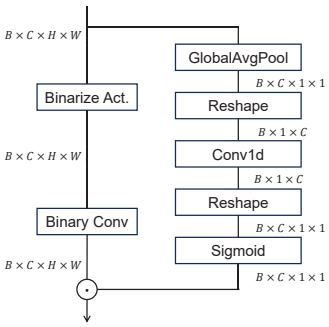


Fig. 7: The proposed channel-wise re-scaling method for CNN-based SR network.

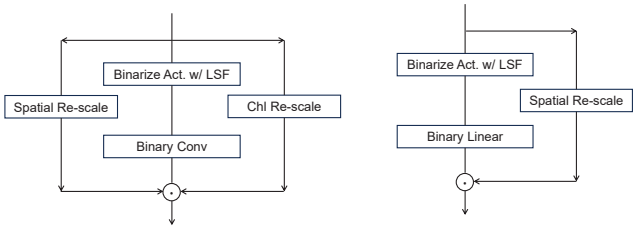
$$A \otimes W \approx (\mathcal{B}_1(A) \otimes \mathcal{B}_2(W)) \odot S(A) \tag{4}$$

where A and W are the FP activations and weights, $\mathcal{B}_1(\cdot)$ and $\mathcal{B}_2(\cdot)$ denote the binarization function for activations and weights respectively, $S(A)$ is our spatial re-scaling method, $\otimes$ denotes the binary convolution or multiplication operation, and $\odot$ denotes the broadcast element-wise multiplication.

C. Channel-wise Re-scaling

To capture the channel-to-channel variation in CNN-based SR networks, we propose the channel-wise re-scaling method in Fig, 7. Note that Transformer-based SR networks do not have channel-to-channel variation due to LN. We use the FP activations before binarization as input since it contains rich information. Then, a global average pooling layer is applied to aggregate the spatial information. Afterward, we capture the inter-channel information with the Conv1d layer and derive the channel-wise scaling factor through a sigmoid function. Our channel-wise re-scaling method can be formulated as:

$$A \otimes W \approx (\mathcal{B}_1(A) \otimes \mathcal{B}_2(W)) \odot C(A) \tag{5}$$



(a) (b)

Fig. 8: The binary (a) convolution and (b) linear layer integrated with SCALES. LSF stands for the layer-wise scaling factor.

where $C(A)$ is our channel-wise re-scaling method. The kernel size of the Conv1d layer is set to 5, which we find have better performance empirically.

Previous work [18] also introduced a channel re-scaling module for image classification BNN. However, our method differs from theirs in the way of generating the channel-wise scaling factor. They adopt a GlobalAvgPool-Linear-ReLU-Linear-Sigmoid structure, which introduces large parameter overhead. Our method only have k FP parameters, which is the kernel size of the Conv1d layer. While the Lineaer-ReLU-Linear structure in [18] introduces $2C^2/r$ FP parameters, where r is the compression ratio, which are $2C^2/rk$ times larger than ours. The ratio will reach 1638 when r is 16, C is 256, and k is 5, which are the typical values.

Combining the aforementioned methods we achieve our proposed binarization method, SCALES. Fig. 8 shows the binary convolution and linear layer equipped with SCALES. For convolution, we also incorporate a skip connection following [11], [17]. The binary convolution and linear layers can serve as a drop-in replacement for various SR network architectures, which enable the binarized SR network to efficiently capture the important pixel-to-pixel, channel-to-channel, layer-to-layer, and image-to-image variations.

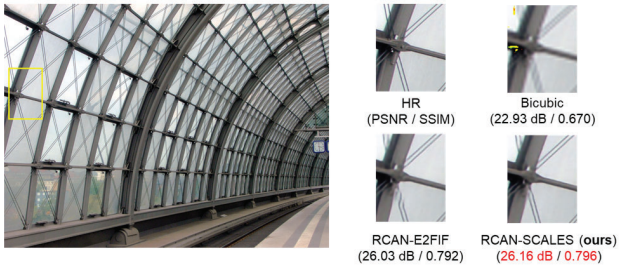
V. Experiments

A. Network Architectures

We evaluate our proposed method SCALES on different SR network architectures. For CNN-based SR networks, we choose SRResNet [1], EDSR [2], RDN [3], and RCAN [14]. For Transformer-based SR networks, we choose SwinIR (Lightweight) [6] and HAT [7]. Following existing works, the head and tail modules are not binarized.

B. Experimental Settings

We train all the models on the training set of DIV2K [31]. For evaluation, we use four standard benchmark datasets including Set5 [32], Set14 [33], B100 [34] and Urban100 [35]. For evaluation metrics, we use PSNR and SSIM [36] over the Y channel of transformed YCbCr space. We choose L1 loss between the SR image and the HR image as our loss function. Input patch size is set to 48×48. The batch size is set to 16. We use ADAM optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We train our models for 300 epochs from scratch.



(a) Visual results from Urban100 at $\times 4$ scale on RCAN architecture

(b) Visual results from Set14 at $\times 2$ scale on EDSR architecture

Fig. 9: Visual comparison of SCALES and the prior art method.

The learning rate is initialized as 2×10^{-4} and halved every 200 epochs.

C. Quantitative and Qualitative Results

We evaluate SCALES on different network architectures on four benchmark datasets across different scales. Due to page limitation, we only show the results on SRResNet, SwinIR, and HAT at $\times 2$ and $\times 4$ scale. For CNN-based SR networks, as shown in Table III, SCALES outperforms the other methods. It surpasses the prior art method E2FIF [11], by 0.22dB and 0.19dB on Urban100 at $\times 2$ and $\times 4$ scale, respectively. For Transformer-based SR networks, since there is no existing research on binarization, we build the baseline model leveraging the binarization method proposed in BiBERT [21]. We also try the method in BiViT [22], but find it less effective than BiBERT [21], thus we choose the better one as our baseline. As shown in Table IV, our proposed method SCALES significantly surpasses the baseline. For example, on SwinIR, SCALES achieves 1.39dB improvement over the baseline on Set5 at $\times 2$ scale. On HAT, the improvement with our method is even larger, i.e., 1.94$\sim$4.31dB across four datasets at $\times 4$ scale. Through our method, we achieve the first accurate binary SR Transformer.

Qualitative results are shown in Figure 9. We can observe that SCALES can alleviate the blurring artifacts and reconstruct clearer images. What's more, SCALES produces more faithful results to the ground truth, compared to the prior art method E2FIF. For example, in Fig. 9a, SCALES has no distortion and in Fig 9b, SCALES generates the stripes with correct directions while E2FIF fails to achieve this.

D. Ablation Study

We evaluate the effect of the proposed individual components on SRResNet in Table V. First, using layer-wise scaling factor

TABLE III: Comparison of different methods on CNN-based SR network (SRResNet).

Method	Scale	Params	OPs	Set5		Set14		B100		Urban100	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRResNet-FP	x2	1517K	913.8G	37.76	0.958	33.27	0.914	31.95	0.895	31.28	0.919
Bicubic	x2	-	-	33.66	0.930	30.24	0.869	29.56	0.843	26.88	0.840
SRResNet-BAM	x2	37K	28.5G	37.21	0.956	32.74	0.910	31.60	0.891	30.20	0.906
SRResNet-BTM	x2	35K	25.8G	37.22	0.957	32.93	0.912	31.77	0.894	30.79	0.914
SRResNet-E2FIF	x2	35K	25.8G	37.50	**0.958**	32.96	0.911	31.79	0.894	30.73	0.913
SRResNet-SCALES (ours)	x2	34K	24.5G	**37.56**	**0.958**	**33.10**	**0.912**	**31.83**	**0.895**	**30.95**	**0.915**
SRResNet-FP	x4	1517K	228.5G	31.76	0.888	28.25	0.773	27.38	0.727	25.54	0.767
Bicubic	x4	-	-	28.42	0.810	26.00	0.703	25.96	0.668	23.14	0.658
SRResNet-BAM	x4	37K	7.1G	31.24	0.878	27.97	0.765	27.15	0.719	24.95	0.745
SRResNet-BTM	x4	35K	6.4G	31.25	0.878	27.94	0.765	27.18	0.720	25.01	0.748
SRResNet-E2FIF	x4	35K	6.4G	31.33	0.880	27.93	0.766	27.20	0.723	25.08	0.750
SRResNet-SCALES (ours)	x4	34K	6.1G	**31.54**	**0.883**	**28.15**	**0.770**	**27.28**	**0.726**	**25.27**	**0.757**

TABLE IV: Comparison of different methods on Transformer-based SR network (SwinIR and HAT).

Method	Scale	Params	OPs	Set5		Set14		B100		Urban100	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SwinIR-FP	x2	878K	391.2G	38.14	0.961	33.86	0.921	32.31	0.901	32.76	0.934
SwinIR-BiBERT	x2	66K	12.5G	35.58	0.947	31.79	0.900	30.80	0.880	28.34	0.877
SwinIR-SCALES (ours)	x2	73K	15.3G	**36.97**	**0.956**	**32.53**	**0.908**	**31.39**	**0.889**	**29.56**	**0.897**
SwinIR-FP	x4	897K	99.2G	32.44	0.898	28.77	0.786	27.69	0.741	26.47	0.798
SwinIR-BiBERT	x4	86K	3.2G	29.52	0.835	26.80	0.734	26.50	0.697	23.77	0.690
SwinIR-SCALES (ours)	x4	93K	3.9G	**29.96**	**0.849**	**27.13**	**0.743**	**26.67**	**0.704**	**24.06**	**0.704**
HAT-FP	x2	20.44M	807.6G	38.73	0.964	35.13	0.928	32.69	0.906	34.81	0.949
HAT-BiBERT	x2	0.86M	25.8G	28.29	0.793	26.46	0.722	26.46	0.699	24.13	0.698
HAT-SCALES (ours)	x2	0.91M	35.9G	**37.34**	**0.958**	**32.97**	**0.912**	**31.76**	**0.894**	**30.61**	**0.912**
HAT-FP	x4	20.80M	204.8G	33.18	0.907	29.38	0.800	28.05	0.753	28.37	0.845
HAT-BiBERT	x4	1.01M	6.6G	26.92	0.774	25.02	0.671	25.23	0.645	22.65	0.639
HAT-SCALES (ours)	x4	1.06M	9.3G	**31.23**	**0.881**	**27.96**	**0.766**	**27.17**	**0.722**	**24.98**	**0.747**

TABLE V: The effect of different components in SCALES. OPs are calculated based on the 128×128 input image. Note that the result of E2FIF is different from the original paper because we train all the model with RGB input instead of YCbCr input.

Method	OPs	Set5		Urban100	
		PSNR	SSIM	PSNR	SSIM
SRResNet-E2FIF	1.83G	31.27	0.880	25.07	0.748
LSF	1.56G	31.30	0.880	25.09	0.751
LSF + chl. re-scale	1.63G	31.42	0.880	25.14	0.753
LSF + spatial re-scale	1.67G	31.48	0.882	25.24	0.756
SCALES	1.74G	31.54	0.883	25.27	0.757

(LSF) already outperforms the prior art method E2FIF with fewer operations. The computation cost is reduced because we remove BN in SRResNet-E2FIF. The proposed channel-wise re-scaling method further improves PSNR by 0.12dB and 0.05dB on Set5 and Urban100, respectively, only increasing 4% operations. The spatial re-scaling method further achieves 0.18dB and 0.15dB improvement on the two datasets. The increase in the number of parameters is negligible for all the components.

E. Deployment Efficiency

We compare the memory and computation cost in Table III and IV. The number of parameters and operations are calculated following [17], [37]: $OPs = OPs^f + OPs^b/64, Param = Param^f + Param^b/32$. We evaluate OPs on a 1280×720 HR image. In Table III, SCALES has the smallest number of parameters and operations. Compared with the prior art method E2FIF, SCALES has better performance with 1K parameters and 0.3G operations reduction due to the removal of BN. In Table IV, SCALES has approximately $10\times$ and $20\times$ parameter reduction compared to SwinIR-FP and HAT-FP, respectively. Compared to the baseline, SCALES only introduces negligible parameters while having significantly better performance.

TABLE VI: Inference latency on mobile phone.

SRResNet	OPs	Params	Latency	Set14		B100	
				PSNR	SSIM	PSNR	SSIM
FP SRResNet	64.98G	1.52M	1649 ms	28.25	0.773	27.38	0.727
E2FIF	1.83G	0.03M	197 ms	27.93	0.766	27.20	0.723
SCALES (chl=64)	1.74G	0.03M	237 ms	28.15	0.770	27.28	0.726
SCALES (chl=40)	0.83G	0.02M	166 ms	28.02	0.767	27.18	0.722

We also benchmark the latency of our method on Redmi K40S phone with a Qualcomm Snapdragon 870 SoC using Larq, an open-source library for deploying BNNs. We report the average latency of 100 times inference using a single thread. As shown in Table VI, with our proposed SCALES (with the number of channels equal to 40), we can achieve $9.9\times$ speedup compared to the FP counterpart, and $1.2\times$ speedup compared to the prior art method E2FIF with on-par performance.

VI. Conclusion

In this paper, we propose an effective binarization method SCALES for both CNN-based and Transformer-based image SR networks, based on our observation that activations in the SR network exhibit large pixel-to-pixel, channel-to-channel, layer-to-layer, and image-to-image variation. With SCALES, we improve the performance of binary CNN-based networks e.g., 0.2dB over the prior art method, with fewer parameters and operations. We also achieve an accurate binary Transformer-based network for the first time, attaining more than 1dB improvement over the baseline method.

REFERENCES

[1] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[2] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[3] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2472–2481.

[4] C. Tian, Y. Yuan, S. Zhang, C.-W. Lin, W. Zuo, and D. Zhang, "Image super-resolution with an enhanced group convolutional neural network," *Neural Networks*, vol. 153, pp. 373–385, 2022.

[5] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 299–12 310.

[6] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1833–1844.

[7] X. Chen, X. Wang, J. Zhou, Y. Qiao, and C. Dong, "Activating more pixels in image super-resolution transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 22 367–22 377.

[8] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV*. Springer, 2016, pp. 525–542.

[9] J. Xin, N. Wang, X. Jiang, J. Li, H. Huang, and X. Gao, "Binarized neural network for single image super resolution," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 2020, pp. 91–107.

[10] X. Jiang, N. Wang, J. Xin, K. Li, X. Yang, and X. Gao, "Training binary neural network without batch normalization for image super-resolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1700–1707.

[11] Z. Lang, C. Song, L. Zhang, and W. Wei, "E2fif: Push the limit of binarized deep imagery super-resolution using end-to-end full-precision information flow," *arXiv preprint arXiv:2207.06893*, 2022.

[12] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[13] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[14] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 286–301.

[15] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," *arXiv preprint arXiv:1903.10082*, 2019.

[16] X. Lin, C. Zhao, and W. Pan, "Towards accurate binary convolutional neural network," *arXiv preprint arXiv:1711.11294*, 2017.

[17] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K.-T. Cheng, "Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 722–737.

[18] B. Martinez, J. Yang, A. Bulat, and G. Tzimiropoulos, "Training binary neural networks with real-to-binary convolutions," *arXiv preprint arXiv:2003.11535*, 2020.

[19] Z. Liu, Z. Shen, M. Savvides, and K.-T. Cheng, "Reactnet: Towards precise binary neural network with generalized activation functions," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 143–159.

[20] Z. Tu, X. Chen, P. Ren, and Y. Wang, "Adabin: Improving binary neural networks with adaptive binary sets," in *European conference on computer vision*. Springer, 2022, pp. 379–395.

[21] H. Bai, W. Zhang, L. Hou, L. Shang, J. Jin, X. Jiang, Q. Liu, M. Lyu, and I. King, "Binarybert: Pushing the limit of bert quantization," *arXiv preprint arXiv:2012.15701*, 2020.

[22] Y. He, Z. Lou, L. Zhang, J. Liu, W. Wu, H. Zhou, and B. Zhuang, "Bivit: Extremely compressed binary vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 5651–5663.

[23] Y. Ma, H. Xiong, Z. Hu, and L. Ma, "Efficient super resolution using binarized neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[24] K. Li, N. Wang, J. Xin, X. Jiang, J. Li, X. Gao, K. Han, and Y. Wang, "Local means binary networks for image super-resolution," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[25] C. Hong, H. Kim, S. Baik, J. Oh, and K. M. Lee, "Daq: Channel-wise distribution-aware quantization for deep image super-resolution networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2675–2684.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[28] A. Bulat and G. Tzimiropoulos, "Xnor-net++: Improved binary neural networks," *arXiv preprint arXiv:1909.13863*, 2019.

[29] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "Smoothquant: Accurate and efficient post-training quantization for large language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 38 087–38 099.

[30] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[31] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 114–125.

[32] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[33] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*. Springer, 2012, pp. 711–730.

[34] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2. IEEE, 2001, pp. 416–423.

[35] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5197–5206.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[37] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.