

# DLGSANet: Lightweight Dynamic Local and Global Self-Attention Network for Image Super-Resolution

Xiang Li, Jiangxin Dong, Jinhui Tang\*, and Jinshan Pan\*

School of Computer Science and Engineering, Nanjing University of Science and Technology

## Abstract

We propose an effective lightweight dynamic local and global self-attention network (DLGSANet) to solve image super-resolution. Our method explores the properties of Transformers while having low computational costs. Motivated by the network designs of Transformers, we develop a simple yet effective multi-head dynamic local self-attention (MHDLSA) module to extract local features efficiently. In addition, we note that existing Transformers usually explore all similarities of the tokens between the queries and keys for the feature aggregation. However, using all the similarities does not effectively facilitate the high-resolution image reconstruction as not all the tokens from the queries are relevant to those in keys. To overcome this problem, we develop a sparse global self-attention (SparseGSA) module to select the most useful similarity values so that the most useful global features can be better utilized for image reconstruction. We develop a hybrid dynamic-Transformer block (HDTB) that integrates the MHDLSA and SparseGSA for both local and global feature exploration. To ease the network training, we formulate the HDTBs into a residual hybrid dynamic-Transformer group (RHDTG). By embedding the RHDTGs into an end-to-end trainable network, we show that the proposed method has fewer network parameters and lower computational costs while achieving competitive performance against state-of-the-art ones in terms of accuracy. More information is available at <https://neonleexiang.github.io/DLGSANet/>.

## 1. Introduction

Single image super-resolution (SISR) aims to find a solution to the issue of reconstructing a high-resolution image from a low-resolution one so that the high-resolution image can be better displayed on high-definition devices. In order to produce high-resolution images, classical approaches, e.g., bicubic and bilinear, employ interpolation processes

\*Corresponding author

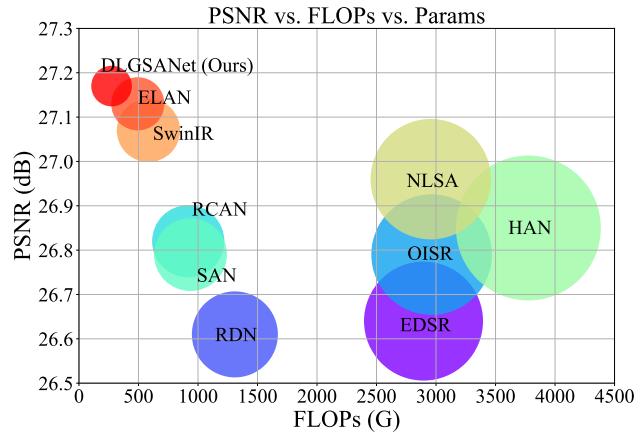


Figure 1. Image super-resolution comparisons ( $\times 4$ ) in terms of accuracy, network parameters, and floating point operations (FLOPs) from the Urban100 dataset. The area of each circle denotes the number of network parameters. Our model (DLGSANet) achieves comparable performance while having fewer network parameters (< 5M) and lower FLOPs.

to complement the surrounding pixel values. Convolutional neural network (CNN)-based approaches such as [7, 8, 15, 21, 33] tackle the image super-resolution challenge, generating better super-resolved images than those of conventional approaches. These CNN-based approaches have greatly advanced the progress of SISR.

Furthermore, several follow-up studies, such as [26, 27, 11], progressively start to develop larger and deeper CNN models for better learning capacity. Although the quality of the super-resolved images is largely improved, the computational costs of those approaches is quite expensive due to the large number of network parameters and calculations (e.g., more than 60M in network parameters and 3000G in FLOPs), which limits their real-world applications. Thus, there is a great need to develop a lightweight and efficient model to solve SISR.

As Vision Transformers (ViTs) [9] can model global contexts while having fewer network parameters, a recent method [4] applies them to SISR and achieves better results

in terms of accuracy and network parameters compared to the CNN-based ones. However, as the original ViTs are computationally expensive, the shifted window scheme has been adopted in [22]. Although the self-attention by the shifted window scheme is capable of extracting local features, discontinuous windows limit the ability to model local features within each window. Moreover, the window-based methods are unable to aggregate information outside of the window, which leads to limited ability for modeling global information.

To better explore global features while reducing the computational costs, several approaches, e.g., [30], develop transposed attentions that compute the self-attention along the number of features. We note that these transformer-based methods usually use all the similarity values in the self-attention for feature aggregation. However, as not all the tokens from the queries are relevant to those in keys, using all similarities does not effectively facilitate the high-resolution image reconstruction. Thus, it is of great interest to develop a method to explore the properties of Transformers for both better local and global feature exploration while reducing the computational costs for high-quality, high-resolution image reconstruction.

In this paper, we propose an effective lightweight dynamic local and global self-attention network (DLGSANet) to solve SISR efficiently. To alleviate the problem caused by the discontinuous windows, we first develop a simple yet effective multi-head dynamic local self-attention (MHDL SA) module. The MHDL SA is motivated by the network designs of Transformers and can dynamically explore the local self-attention based on a fully CNN model to better extract local features. As not all the tokens from the queries are relevant to those in keys, using all similarities does not effectively facilitate the high-resolution image reconstruction. To overcome this problem, we develop a sparse global self-attention (SparseGSA) module to select the most useful similarity values for feature aggregation. We propose a hybrid dynamic-Transformer block (HDTB) that integrates the MHDL SA and SparseGSA to explore both local and global features for high-resolution image reconstruction. We further develop a residual hybrid dynamic-Transformer group (RHDTG) that stacks the HDTB based on the residual learning. We formulate the RHDTGs into an end-to-end trainable network, named DLGSANet, to solve SISR. Figure 1 shows that the proposed DLGSANet model achieves comparable performance with fewer network parameters and lower computational costs.

The main contributions of this work are summarized as follows:

- We propose a lightweight SISR model, called DLGSANet, to solve the SISR problem efficiently and effectively. Our analysis shows that the proposed model has fewer network parameters (< 5M) and needs lower

computational costs while generating competitive performance.

- We propose a simple yet effective multi-head dynamic local self-attention (MHDL SA) module to extract local features dynamically.
- We develop an effective sparse global self-attention module (SparseGSA) to generate better self-attention for global feature exploration.

## 2. Related Work

**Conventional CNNs for SR.** SRCNN [7] firstly introduces an effective end-to-end trainable CNN to solve the image super-resolution (SR) task. Then, VDSR [15] further improves the performance of CNNs by deepening the network and introducing residual learning, which leads to the emergence of a growing number of CNNs [8, 16, 29, 18] for SR tasks. EDSR [21] further improves PSNR results significantly by removing the unnecessary BatchNormal [14] layers. Additionally, RCAN [33] uses a channel attention mechanism to enable the network’s capability of efficient feature aggregation, allowing the network to perform better with a deeper network. Then, an increasing number of models, including SAN [6], NLSA [26], and HAN [27], propose a variety of attention mechanisms along spatial or channel dimensions. Although these models produce significant results, a large number of parameters are required to build the network for better feature aggregation.

**Efficient SR.** Instead of aggregating on a single picture of fixed resolution, FSRCNN [8] uses a post-upsampling approach to reduce FLOPs expenses. To increase efficiency, CARN [1] applies group convolution and a cascade method to a residual network. While IMDN [13] further reduces the parameters with information multi-distillation blocks. LatticeNet [13] further improves the PSNR results with lattice blocks and with comparable parameter numbers and low FLOPs expenses. Although these models are lightweight and efficient, the quality of the restored high-resolution images is not good compared to the large SR models.

**Transformer-based methods for SR.** Transformer-based methods [4, 20] are proposed to solve image restoration tasks such as SR tasks. SwinIR [20] uses the window-based attention mechanism to solve image SR and outperforms the CNN-based method in terms of accuracy and model complexity. ELAN [32] proposes a share attention technique to speed up the calculation in its group multi-head self-attention (GMSA). On the other hand, with comparable parameter numbers and computational costs, SwinIR-light [20] surpasses state-of-the-art methods [13, 24, 1, 19]. ELAN-light [32] further reduces the inference time.

Different from existing methods, we propose a lightweight DLGSANet which needs lower computational costs for better image SR.

### 3. Proposed Method

The proposed lightweight dynamic local and global self-attention network (DLGSANet) mainly contains a shallow feature extraction module, six residual hybrid dynamic-Transformer groups (RHDTGs) for both local and global feature extraction, and a high-resolution image reconstruction module. The shallow feature extraction uses a convolutional layer with a filter size of  $3 \times 3$  pixels to extract features from the input low-resolution image. Each RHDTG takes the hybrid dynamic-Transformer block (HDTB) as the basic module. Moreover, the HDTB contains the multi-head dynamic local self-attention (MHDLSA) and the sparse global self-attention (SparseGSA). The high-resolution image reconstruction module contains a convolutional layer with a filter size of  $3 \times 3$  pixels, followed by a PixelShuffle [28] operation for upsampling.

Figure 2 shows the overview of the proposed DLGSANet for SISR. In the following, we explain the details of the MHDLSA, SparseGSA, and RHDTG.

#### 3.1. Multi-head dynamic local self-attention

We note that window-based self-attention methods alleviate the huge computational costs of Transformers and achieve decent performance in SISR, as shown in [20] and [32]. However, the split windows cannot effectively extract features continuously and are unable to aggregate the information outside of the windows. Although the shifted windows are able to model the long-distance connections of the features in different windows, they lead to additional computational costs.

To overcome this problem, we propose a simple yet effective multi-head dynamic local self-attention (MHDLSA) based on the network designs of Transformers to extract local features effectively and efficiently. The proposed MHDLSA first estimates spatial-variant filters to explore the local features dynamically. Then, we use the estimated filters as the dynamic local attention and apply them to the input features for better local feature aggregation. Finally, similar to the Transformers that use a feed-forward network to improve feature representation, we apply a gated feed-forward network by [30] to the aggregated features for better performance.

Specifically, given a feature  $\mathbf{Y}_{in} \in \mathbb{R}^{H \times W \times C}$  generated by a layer norm followed by a  $1 \times 1$  convolution, we first develop a dynamic weight generation network without any normalization layer and non-linear activation. To ensure the generated dynamic weight better models the local information, we further use a depth-wise convolutional layer in the dynamic weight generation network as the depth-wise convolutional operation is able to model local attentions [23].

The proposed dynamic weight generation is achieved by:

$$\begin{aligned}\mathbf{Y} &= \text{DConv}_{7 \times 7}(\text{Conv}_{1 \times 1}(\mathbf{Y}_{in})), \mathbf{Y} \in \mathbb{R}^{H \times W \times \gamma C} \\ \mathbf{Y}_{out} &= \text{Conv}_{1 \times 1}(\mathbf{Y}), \mathbf{Y}_{out} \in \mathbb{R}^{H \times W \times G \times K^2} \\ \mathbf{W}(x) &= \mathcal{R}(\mathbf{Y}_{out}), \mathbf{W}(x) \in \mathbb{R}^{G \times K \times K}\end{aligned}\quad (1)$$

where  $\gamma$  denotes a squeezing factor;  $\text{DConv}_{7 \times 7}$  denotes a depth-wise convolution with filter size of  $7 \times 7$  pixels;  $\text{Conv}_{1 \times 1}$  denotes a convolution with a filter size of  $1 \times 1$  pixel;  $\mathcal{R}$  denotes a reshaping function;  $x$  denotes the pixel index. Each pixel has a correlated  $K \times K$  dynamic kernel for dynamic convolution.

With the generated pixel-wise weight  $\mathbf{W}$ , we obtain the aggregated feature by:

$$\hat{\mathbf{X}}^l = \mathbf{W} \circledast \mathbf{Y}_{in}, \quad (2)$$

where  $\circledast$  denotes the Dynamic convolution operation with weight-sharing mechanism for each channel [10].

The detailed network of the dynamic weight generation is shown in Figure 2. Similar to the multi-head self-attention methods [30, 20, 22], we divide the number of feature channels into  $G$  heads and learn separate dynamic weights in parallel.

As the feed-forward network is widely used in Transformers for the better feature representation ability, we further apply an improved feed-forward network by [30] to the aggregated feature  $\hat{\mathbf{X}}$ :

$$\mathbf{X}^l = FFN(\hat{\mathbf{X}}^l), \quad (3)$$

where  $FFN(\cdot)$  denotes a feed-forward network and its network details are included in Figure 2.

#### 3.2. Sparse global self-attention

Although the MHDLSA is able to estimate features dynamically, it is less effective to model global features as the generated dynamic filters are based on fully convolutional operations. Transformer-based methods are able to explore global features. However, they are usually computationally expensive. Recent method [30] develops an efficient transposed self-attention that is estimated along feature channel dimension. Although it is efficient, the scaled dot-production attention is still generated by a softmax normalization. We note that the softmax normalization will keep all the similarities between the tokens from the query and key. However, not all the tokens from the queries are relevant to those in keys. Using the softmax normalization to generate self-attention would affect the following feature aggregation. To overcome this problem, we propose a simple yet effective sparse global self-attention module. As the ReLU is an effective activation function that can remove negative features while keeping the positive ones, we use

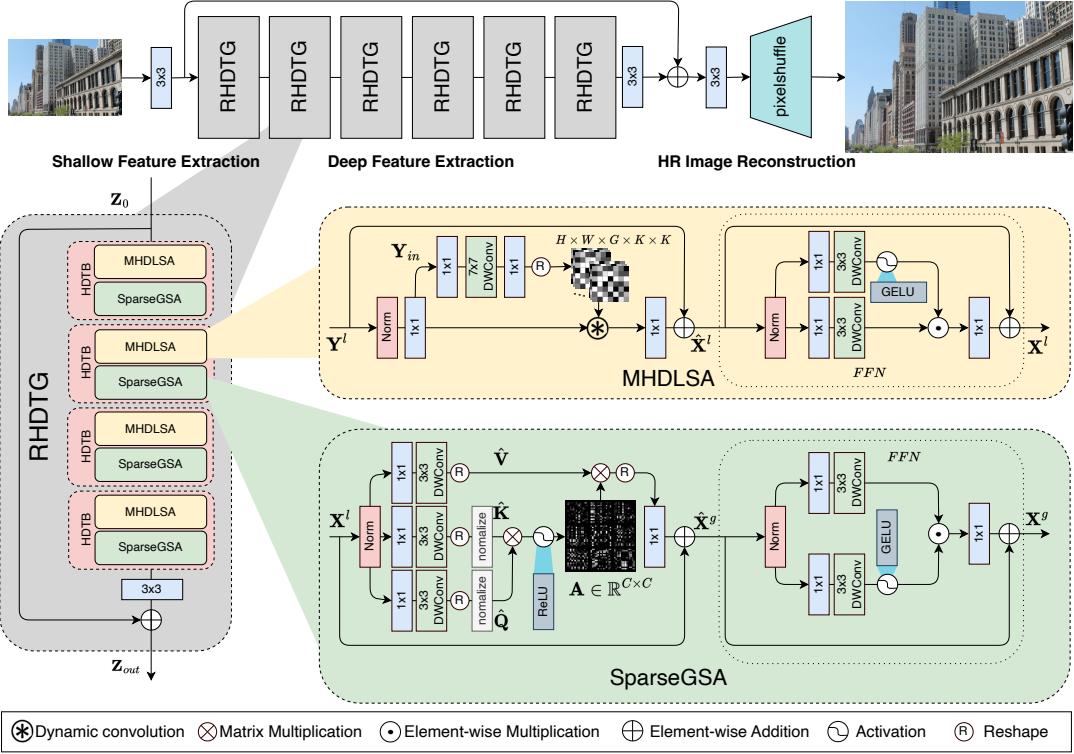


Figure 2. Network architecture of the proposed DLGSANet. It mainly contains a shallow feature extraction module, six residual hybrid dynamic-Transformer groups (RHDTGs) for both local and global feature extraction, and a high-resolution image reconstruction module.

the ReLU to keep the most useful attention for feature aggregation.

Given a normalized feature  $\mathbf{X}^l \in \mathbb{R}^{H \times W \times C}$  generated by the MHDLA module, we first use a  $1 \times 1$  convolution followed by a  $3 \times 3$  depth-wise convolution to generate the query  $\mathbf{Q} \in \mathbb{R}^{H \times W \times C}$ , key  $\mathbf{K} \in \mathbb{R}^{H \times W \times C}$ , and  $\mathbf{V} \in \mathbb{R}^{H \times W \times C}$ . Based on [30], we respectively apply a reshaping function to the query  $\mathbf{Q}$ , key  $\mathbf{K}$ , and value  $\mathbf{V}$  and obtain  $\hat{\mathbf{Q}} \in \mathbb{R}^{HW \times C}$ ,  $\hat{\mathbf{K}} \in \mathbb{R}^{HW \times C}$ , and  $\hat{\mathbf{V}} \in \mathbb{R}^{HW \times C}$ . To keep the most useful attention for feature aggregation, we compute the self-attention by:

$$\mathbf{A} = \text{ReLU} \left( \frac{\hat{\mathbf{Q}}^\top \hat{\mathbf{K}}}{\alpha} \right), \mathbf{A} \in \mathbb{R}^{C \times C} \quad (4)$$

where  $\alpha$  is a learnable parameter. Here we use the ReLU to keep the most useful attention as it is simple while can generate better results (see analysis in Section 5). With the estimated attention  $\mathbf{A}$ , we use the same operation by [30] to generate the output aggregated feature  $\hat{\mathbf{X}}^g \in \mathbb{R}^{H \times W \times C}$ . Then the improved feed-forward network by [30] is applied to  $\hat{\mathbf{X}}^g$  to generate the output (i.e.,  $\mathbf{X}^g$  in Figure 2). The network details of the sparse global self-attention module are shown in Figure 2.

We note that using Eq. (4) leads to a sparse self-attention (SparseGSA) that can keep the most useful features for

high-resolution image reconstruction. The effectiveness of the proposed SparseGSA will be detailed in Section 5.

### 3.3. Residual hybrid dynamic-Transformer group

By exploring the MHDLA and SparseGSA, we develop a hybrid dynamic-transformer block (HDTB) that contains the MHDLA and SparseGSA for local and global feature estimations. We directly cascade the MHDLA and SparseGSA to construct our HDTB so that the SparseGSA can fuse sparse global information on the features of the fusion of local information produced by the MHDLA in a simple yet more effective way. To reduce the training difficulty, we embed the HDTB into a residual learning framework, which leads to a hybrid dynamic-Transformer group (RHDTG). Specifically, given the input feature  $\mathbf{Z}_0$ , the proposed RHDTG is achieved by:

$$\begin{aligned} \mathbf{Z}_i &= \mathcal{M}_i(\mathbf{Z}_{i-1}), i = 1, 2, 3, \dots, N, \\ \mathbf{Z}_{out} &= \text{Conv}_{3 \times 3}(\mathbf{Z}_N) + \mathbf{Z}_0, \end{aligned} \quad (5)$$

where  $\mathcal{M}_i$  denotes the  $i$ -th HDTB.

Finally, we formulate the proposed RHDTG into an end-to-end deep CNN model to solve SISR. The whole network is shown in Figure 2.

Table 1. Quantitative evaluations of the proposed DLGSANet against state-of-the-art methods on commonly used SISR benchmark datasets. #Params means the number of the network parameters. #FLOPs denotes the number of the FLOPs, which are calculated on images with an upscaled spatial resolution of  $1280 \times 720$  pixels. Best and second best results are marked in red and blue colors.

Scale	Method	#Params/(M)	#FLOPs/(G)	Set5	Set14	B100	Urban100	Manga109
$\times 2$	EDSR [21]	40.73	9387	38.11/0.9602	33.92/0.9195	32.32/0.9013	32.93/0.9351	39.10/0.9773
	RDN [34]	22.12	5098	38.24/0.9614	34.01/0.9212	32.34/0.9017	32.89/0.9353	39.18/0.9780
	RCAN [33]	15.44	3530	38.27/0.9614	34.12/0.9216	32.41/0.9027	33.34/0.9384	39.44/0.9786
	SAN [6]	15.86	3050	38.31/0.9620	34.07/0.9213	32.42/0.9028	33.10/0.9370	39.32/0.9792
	HAN [27]	63.60	14551	38.27/0.9614	34.16/0.9217	32.41/0.9027	33.35/0.9385	39.46/0.9785
	NLSA [26]	41.79	9632	38.34/0.9618	34.08/0.9231	32.43/0.9027	33.42/0.9394	39.59/0.9789
	SwinIR [20]	11.75	2301	38.35/0.9620	34.14/0.9227	32.44/0.9030	33.40/0.9393	39.60/0.9792
	ELAN [32]	8.25	1965	38.36/0.9620	34.20/0.9228	32.45/0.9030	33.44/0.9391	39.62/0.9793
$\times 3$	<b>DLGSANet (Ours)</b>	<b>4.73</b>	<b>1097</b>	38.34/0.9617	34.25/0.9231	32.38/0.9025	33.41/0.9393	39.57/0.9789
	EDSR [21]	43.68	4470	34.65/0.9280	30.52/0.8462	29.25/0.8093	28.80/0.8653	34.17/0.9476
	RDN [34]	22.30	2282	34.71/0.9296	30.57/0.8468	29.26/0.8093	28.80/0.8653	34.13/0.9484
	RCAN [33]	15.62	1586	34.74/0.9299	30.65/0.8482	29.32/0.8111	29.09/0.8702	34.44/0.9499
	SAN [6]	15.89	1620	34.75/0.9300	30.59/0.8476	29.33/0.8112	28.93/0.8671	34.30/0.9494
	HAN [27]	64.34	6534	34.75/0.9299	30.67/0.8483	29.32/0.8110	29.10/0.8705	34.48/0.9500
	NLSA [26]	44.74	4579	34.85/0.9306	30.70/0.8485	29.34/0.8117	29.25/0.8726	34.57/0.9508
	SwinIR [20]	11.93	1026	34.89/0.9312	30.77/0.8503	29.37/0.8124	29.29/0.8744	34.74/0.9518
$\times 4$	ELAN [32]	8.27	874	34.90/0.9313	30.80/0.8504	29.38/0.8124	29.32/0.8745	34.73/0.9517
	<b>DLGSANet (Ours)</b>	<b>4.74</b>	<b>486</b>	<b>34.95/0.9310</b>	<b>30.77/0.8501</b>	<b>29.38/0.8121</b>	<b>29.43/0.8761</b>	<b>34.76/0.9517</b>
	EDSR [21]	43.09	2895	32.46/0.8968	28.80/0.7876	27.71/0.7420	26.64/0.8033	31.02/0.9148
	RDN [34]	22.27	1310	32.47/0.8990	28.81/0.7871	27.72/0.7419	26.61/0.8028	31.00/0.9151
	RCAN [33]	15.59	918	32.63/0.9002	28.87/0.7889	27.77/0.7436	26.82/0.8087	31.22/0.9173
	SAN [6]	15.86	937	32.64/0.9003	28.92/0.7888	27.78/0.7436	26.79/0.8068	31.18/0.9169
	HAN [27]	64.19	3776	32.64/0.9002	28.90/0.7890	27.80/0.7442	26.85/0.8094	31.42/0.9177
	NLSA [26]	44.15	2956	32.59/0.9000	28.87/0.7891	27.78/0.7444	26.96/0.8109	31.27/0.9184
$\times 5$	SwinIR [20]	11.90	584	32.72/0.9021	28.94/0.7914	27.83/0.7459	27.07/0.8164	31.67/0.9226
	ELAN [32]	8.31	494	32.75/0.9022	28.96/0.7914	27.83/0.7459	27.13/0.8167	31.68/0.9226
	<b>DLGSANet (Ours)</b>	<b>4.76</b>	<b>274</b>	<b>32.80/0.9021</b>	<b>28.95/0.7907</b>	<b>27.85/0.7464</b>	<b>27.17/0.8175</b>	<b>31.68/0.9219</b>

## 4. Experimental Results

In this section, we perform both quantitative and qualitative evaluations to demonstrate the effectiveness of the proposed DLGSANet on commonly used benchmark datasets.

### 4.1. Experimental settings

**Datasets.** We adopt the commonly used DIV2K dataset as the training dataset and evaluate our method on the commonly used test datasets, including Set5 [3], Set14 [31], B100 [2], Urban100 [12], and Manga109 [25].

**Implementation details.** In the proposed DLGSANet, we use 6 RHDTGs, where each RHDTG contains 4 HDTBs. The feature channel number is set to be 90, the kernel size of the MHDLSA is set to be 7 and the multi-head number of the HDTB is set to be 6. We also evaluate the proposed DLGSANet in lightweight settings by reducing the numbers of the RHDTG, the HDTB, and the feature channel. When the numbers of the RHDTG, the HDTB, and the feature channel are set to be 3, 3, and 48, respectively, we refer to the DLGSANet as DLGSANet-tiny. When the numbers of the RHDTG, the HDTB, and the feature channel are set to be 4, 3, and 48, respectively, we refer to the DLGSANet as DLGSANet-light. During the training, the mini-batch size is set to be 16. The patch size is set to be  $48 \times 48$  pixels. The initial learning rate is set to be  $5 \times 10^{-4}$  with a multi-step scheduler in 500K iterations. We train our model using

the Adam optimizer [17] with default parameter settings. All the networks are trained and performed using the PyTorch framework on a machine with two NVIDIA GeForce RTX 3090 GPUs. As pointed out by [5], the global attention in image restoration usually has a gap between the training and testing stages, we thus use the test-time local converter (TLC) approach by [5] during the testing stage.

Following the protocols used in existing methods (e.g., [20, 33, 21]), we calculate the PSNR and SSIM scores using the Y channel in the YCbCr color space as quantitative comparisons. Moreover, the FLOPs of each evaluated method are obtained based on upscaled images with a spatial resolution of  $1280 \times 720$  pixels.

### 4.2. Comparison results

We compare the proposed DLGSANet with state-of-the-art methods, including SwinIR [20], ELAN [32], NLSA [26], HAN [27], RCAN [33], and EDSR [21].

**Quantitative evaluations.** Table 1 shows the quantitative evaluation results on the commonly used SR image benchmarks. We note that the proposed DLGSANet performs favorably against state-of-the-art methods in terms of network parameters and FLOPs while generating competitive results. Particularly, compared to conventional CNN-based models, e.g., EDSR [21], the proposed DLGSANet achieves 0.62dB gains on the Urban100 dataset in terms

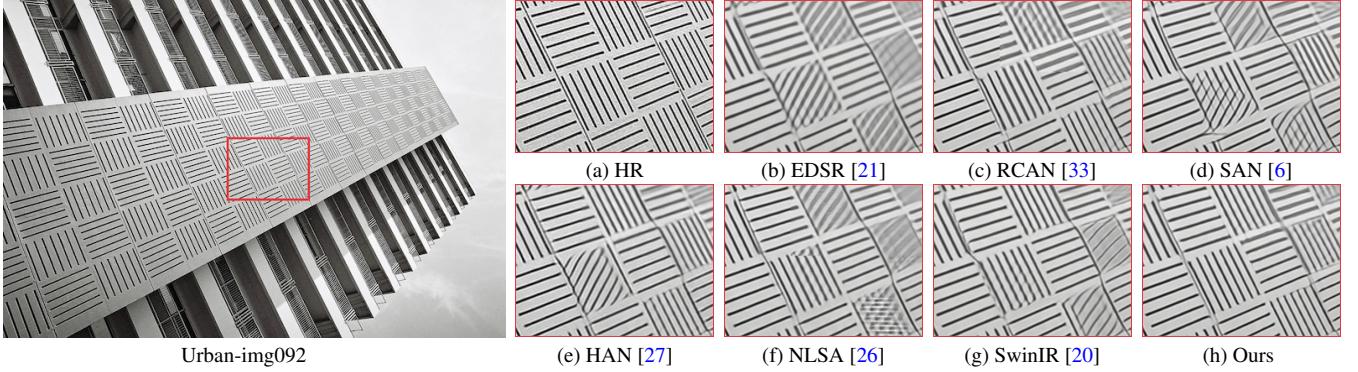


Figure 3. Super-resolution results ( $\times 4$ ) on the “img092” image from the Urban100 dataset. The structures of the stripes are not recovered well by the evaluated methods.

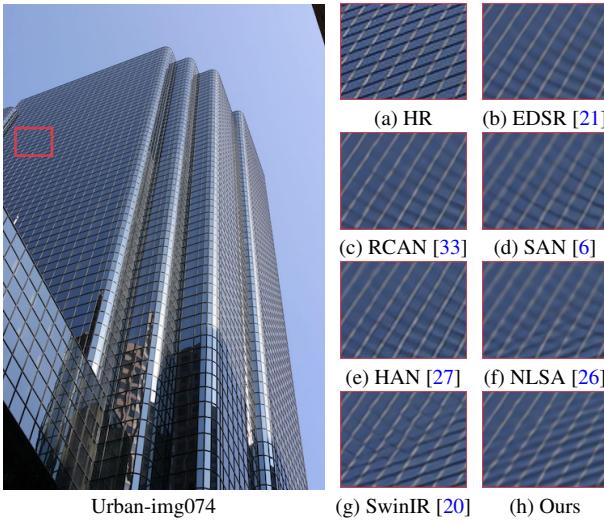


Figure 4. Super-resolution results ( $\times 4$ ) on the “img074” image from the Urban100 dataset. The evaluated methods do not recover the windows of the building well, as shown in (b)-(g).

of PSNR, while the network parameters and FLOPs of the EDSR method are  $\times 10$  times than those of our DLGSANet. Compared to the channel attention-based method [33, 26], our DLGSANet achieves 0.35dB and 0.21dB gains on the Urban100 dataset in terms of PSNR while utilizing  $\times 3$  times and  $\times 10$  times fewer parameters and FLOPs. When compared to the Transformer-based methods, our DLGSANet slightly outperforms the most recent approaches, SwinIR and ELAN. As shown in Table 1, DLGSANet performs better on the Urban100 when the scale factor is  $\times 4$  while our method has fewer network parameters and lower FLOPs than the SwinIR method [20]. We note that the ELAN method [32] outperforms the SwinIR method [20]. However, our method still generates comparable results. More importantly, our method has fewer network parameters and lower FLOPs than the ELAN method [32]. All comparisons presented in Table 1 show that DLGSANet is lightweight and much more efficient than the state-of-the-

art methods.

**Qualitative evaluations.** We compare the visual results of  $\times 4$  super-resolution on the Urban100 dataset between the proposed method and state-of-the-art ones (EDSR [21], RCAN [33], SAN [6], HAN [27], NLSA [26], SwinIR [20]). Figure 3 shows visual comparisons of the evaluated methods. As the typical convolutional layers do not model the locally variant structures, the CNN-based methods do not correct boundaries. The window-based self-attention methods do not effectively aggregate information outside of the windows, which thus affects the quality of the restored image (see Figure 3(g)). In contrast, our DLGSANet explores both local and global information by the MHDLSA and SparseGSA and restores a better image with clear blocks and boundaries, as shown in Figure 3(h).

Figure 4 shows another visual comparison, where our method generates a better super-resolved image than the evaluated methods.

**Comparisons with lightweight models.** We also compare DLGSANet-tiny and DLGSANet-light with the state-of-the-art lightweight SISR models, including EDSR-baseline [21], IMDN [13], LatticNet [24], SwinIR-light [20], and ELAN-light [32]. Table 2 shows that our proposed DLGSANet-tiny and DLGSANet-light perform better than the lightweight state-of-the-art deep models on five datasets. Particularly, the DLGSANet-tiny has the fewest network parameters and the lowest FLOPs. In addition, it is worth mentioning that our DLGSANet-light performs better than ELAN-light (0.21dB gains on  $\times 4$  Manga109) while the DLGSANet-light has similar FLOPs to ELAN-light.

## 5. Ablation Study and Analysis

In this section, we further evaluate the effect of the components in the proposed method and compare the proposed method with baseline models. For fair comparisons, we train all the baseline models using the same settings as the proposed DLGSANet. We use the Urban100 dataset as the test dataset, as it contains a variety of images with various

Table 2. Quantitative evaluations of the lightweight DLGSANet against state-of-the-art methods on commonly used benchmark datasets. Best and second best results are marked in red and blue colors. #Params means the number of the network parameters. #FLOPs denotes the number of the FLOPs which are calculated on images with an upscaled spatial resolution of  $1280 \times 720$  pixels.

Scale	Method	#Params(/K)	#FLOPs(/G)	Set5	Set14	B100	Urban100	Manga109
$\times 2$	EDSR-baseline [21]	1370	316.3	37.99/0.9604	33.57/0.9175	32.16/0.8994	31.98/0.9272	38.54/0.9769
	IMDN [13]	694	158.8	38.00/0.9605	33.63/0.9177	32.19/0.8996	32.17/0.9283	38.88/0.9774
	LatticeNet [24]	756	169.5	38.06/0.9607	33.70/0.9187	32.20/0.8999	32.25/0.9288	/
	SwinIR-light [20]	878	195.6	38.14/ <b>0.9611</b>	33.86/ <b>0.9206</b>	<b>32.31/0.9012</b>	32.76/0.9340	39.12/ <b>0.9783</b>
	ELAN-light [32]	582	<b>168.4</b>	<b>38.17/0.9611</b>	<b>33.94/0.9207</b>	<b>32.30/0.9012</b>	32.76/0.9340	39.11/ <b>0.9782</b>
$\times 3$	<b>DLGSANet-tiny (Ours)</b>	<b>566</b>	<b>128.1</b>	38.16/ <b>0.9611</b>	<b>33.92/0.9202</b>	32.26/0.9007	<b>32.82/0.9343</b>	39.14/0.9777
	<b>DLGSANet-light (Ours)</b>	745	170	<b>38.20/0.9612</b>	33.89/0.9203	<b>32.30/0.9012</b>	<b>32.94/0.9355</b>	<b>39.29/0.9780</b>
	EDSR-baseline [21]	1555	160.2	34.37/0.9270	30.28/0.8417	29.09/0.8052	28.15/0.8527	33.45/0.9439
	IMDN [13]	703	71.5	34.36/0.9270	30.32/0.8417	29.09/0.8046	28.17/0.8519	33.61/0.9445
$\times 4$	LatticeNet [24]	765	76.3	34.40/0.9272	30.32/0.8416	29.10/0.8049	28.19/0.8513	/
	SwinIR-light [20]	886	87.2	34.62/ <b>0.9289</b>	30.54/ <b>0.8463</b>	29.20/0.8082	28.66/0.8624	33.98/0.9478
	ELAN-light [32]	590	75.7	34.61/0.9288	30.55/ <b>0.8463</b>	<b>29.21/0.8081</b>	<b>28.69/0.8624</b>	34.00/0.9478
	<b>DLGSANet-tiny (Ours)</b>	<b>572</b>	<b>56.8</b>	34.63/0.9288	<b>30.57/0.8459</b>	<b>29.21/0.8083</b>	<b>28.69/0.8630</b>	<b>34.10/0.9480</b>
	<b>DLGSANet-light (Ours)</b>	752	<b>75.4</b>	<b>34.70/0.9295</b>	<b>30.58/0.8465</b>	<b>29.24/0.8089</b>	<b>28.83/0.8653</b>	<b>34.16/0.9483</b>

kinds of structural information.

**Effectiveness of the HDTB.** As one of the key components in our DLGSANet, the HDTB fuses both local and global information for better feature aggregation. As the HDTB contains MHDLSA and SparseGSA, we compare the proposed method with two baselines. One baseline is that we use two MHDLSA blocks in HDTB (HDTB<sub>MHDLSA</sub> for short). The other one is that we use two SparseGSA blocks in HDTB (HDTB<sub>SparseGSA</sub> for short). The main reason we use two blocks in the HDTB is to ensure these baseline models have similar network parameters as the proposed network. We train these two baselines using the same settings as the proposed method for fairness. Table 3 shows that only using the MHDLSA generates the results with a PSNR value of 26.88dB and using the SparseGSA generates the results with a PSNR value of 26.86dB. The PSNR values of these two baselines are lower than the HDTB, suggesting the effectiveness of using both MHDLSA and SparseGSA in the HDTB for SISR. Figure 5(b) and (c) show that only using the MHDLSA or the SparseGSA in the HDTB does not restore the structures well. The SparseGSA primarily concentrates on global information fusion while lacking local feature aggregation due to a few layers of depth-wise convolutions. In contrast, using both the MHDLSA and SparseGSA in HDTB leads to a clearer image with finer structural details (see Figure 5(d)).

**Effectiveness of the MHDLSA.** Our MHDLSA approach inherits the property of convolution and can generate dynamic weights for better local feature exploration. To demonstrate the effectiveness of the proposed MHDLSA, we first replace the MHDLSA with the commonly used multi-head window attention (MHSA) in the proposed net-

Table 3. Ablation study w.r.t. the MHDLSA and SparseGSA in the HDTB. The results ( $\times 4$ ) are obtained from the Urban100 dataset.

Model	MHDLSA	SparseGSA	#Param	PSNR
HDTB <sub>MHDLSA</sub>	✓		4.79M	26.88
HDTB <sub>SparseGSA</sub>		✓	4.73M	26.86
HDTB	✓	✓	4.76M	<b>27.17</b>

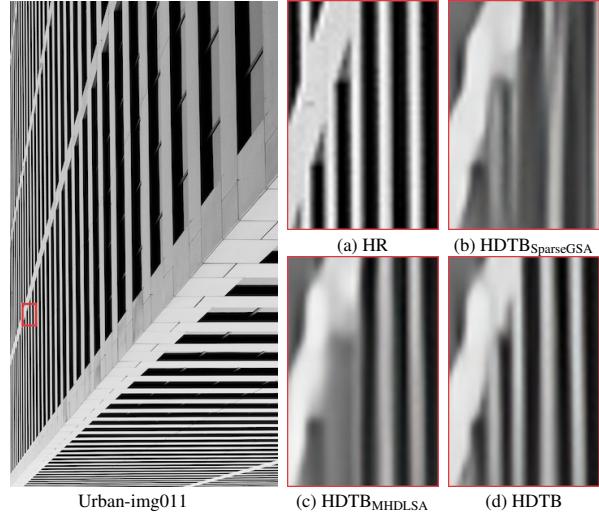


Figure 5. Effect of the MHDLSA and the SparseGSA in the HDTB for SISR. The results ( $\times 4$ ) are obtained from the “img011” image of the Urban100 dataset.

work and train this baseline using the same settings as the proposed network for fair comparisons. Table 4 shows that using the MHDLSA achieves 0.28dB gains in terms of PSNR compared to the method using the MHSA, suggesting the effectiveness of the MHDLSA on SISR.

**Effectiveness of the SparseGSA.** The proposed

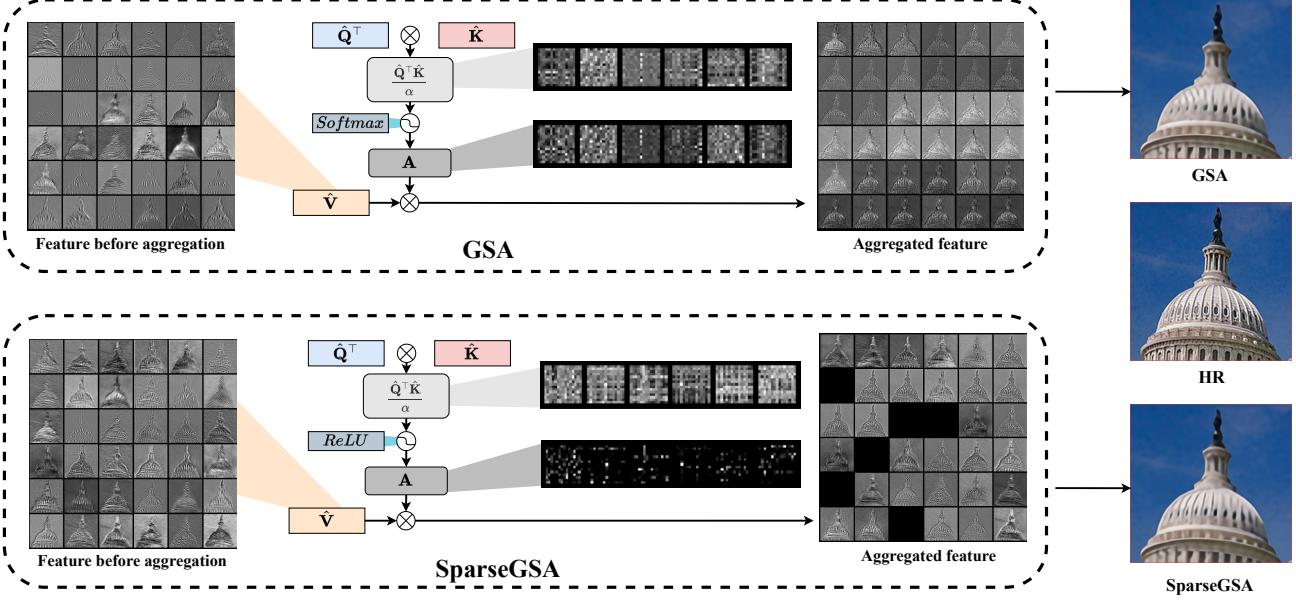


Figure 6. Effect of the SparseGSA on SISR. Using the SparseGSA is able to remove useless self-attention values and thus leads to better features for high-resolution image reconstruction.

Table 4. Effect of the proposed MHDLA. The results ( $\times 4$ ) are obtained from the Urban100 dataset.

Model	MHSA	MHDLA	SparseGSA	#Param	PSNR
w/ MHSA	✓		✓	4.67M	26.89
w/ MHDLA		✓	✓	4.76M	<b>27.17</b>

Table 5. Effectiveness of the proposed SparseGSA. The results ( $\times 4$ ) are obtained from the Urban100 dataset.  $Sparsity_{avg}$  denotes the average sparsity of the attention map of all the images in the Urban100 dataset.

Model	Softmax	ReLU	Sparsity <sub>avg</sub>	#Param	PSNR
GSA	✓		100%	4.76M	27.05
SparseGSA		✓	22.6%	4.76M	<b>27.17</b>

SparseGSA uses the ReLU to remove useless self-attention for better feature aggregation. We demonstrate the effectiveness of the SparseGSA by comparing it with the commonly used method that adopts the softmax operation. Table 5 demonstrates that the SparseGSA outperforms the commonly used method that uses the softmax for self-attention, where the PSNR value of the method using the SparseGSA is 0.12dB higher.

We further show visualization results in Figure 6 to better illustrate the effect of the proposed SparseGSA. As shown in Figure 6, the softmax keeps all the similarities between the tokens and many aggregated features. Extra information cannot be provided by redundant features, which might affect the feature aggregation (as shown in Figure 6, the GSA leads to wrong church dome details). In contrast, the SparseGSA leads to a better result by generating a sparse attention map to keep the features with rich structural details and drop the insignificant features to prevent the effects on subsequent feature aggregations.

To better understand the sparsity of the attention map, we use the Urban100 [12] dataset and calculate the average sparsity of the attention map of all the images in the Urban100 dataset. Given the sparse attention matrix  $\mathbf{A} \in \mathbb{R}^{C \times C}$  defined in the Eq. (4) of Section 3, the Sparsity is achieved by:

$$Sparsity = \frac{\sum_{i=1}^C \sum_{j=1}^C \#(a_{ij} > 0)}{C \times C} \times 100\%, a_{ij} \in \mathbf{A}, \quad (6)$$

where  $\#(\cdot)$  denotes the counting operator.

Table 5 shows that on the Urban100 dataset, GSA uses 100% of the attention values for feature aggregations while SparseGSA uses an average of 22.6% of the attention values. Table 5 and Figure 6 demonstrate that only a few input features with the most useful details are aggregated into each output feature.

We further illustrate the effectiveness of the SparseGSA by comparing it to other activation functions. Table 6 shows that our SparseGSA generates the results with higher PSNR values. Since Sigmoid (GSA<sub>Sigmoid</sub>) and GELU (GSA<sub>GELU</sub>) still keep most of the similarity values of the tokens, they are unable to remove useless self-attentions for the feature aggregation, which thus affects the final results.

Table 6. The effectiveness of the proposed SparseGSA. The results ( $\times 4$ ) are obtained from the Urban100 dataset.

Model	GSA	GSA <sub>Sigmoid</sub>	GSA <sub>GELU</sub>	SparseGSA
PSNR	27.05	27.09	27.11	<b>27.17</b>

**Running time analysis.** We further evaluate the running time of the proposed DLGSANet against the state-of-the-

Table 7. Evaluations of running time (/ms) on NVIDIA GeForce RTX 3090 GPUs.

Type	Model	x2	x3	x4
Lightweight (< 1M)	EDSR-baseline [21]	40	21	15
	IMDN [13]	29	13	8
	LatticNet [24]	36	17	10
	SwinIR-light [20]	340	145	81
	ELAN-light [32]	165	78	46
	<b>DLGSANet-tiny (Ours)</b>	143	66	38
Regular	<b>DLGSANet-light (Ours)</b>	192	88	51
	EDSR [21]	679	344	232
	RCAN [33]	487	220	133
	NLSA [26]	1208	548	343
	SwinIR [20]	1314	528	278
	ELAN [32]	965	422	243
	<b>DLGSANet (Ours)</b>	748	337	187

art methods by using a machine with an NVIDIA GeForce RTX 3090 GPU. We use test images with the upscaled spatial resolution of  $1280 \times 720$  pixels. Table 7 shows that our method, including both the regular model and the lightweight model, is more efficient than the Transformer-based methods.

## 6. Conclusion

We have presented an effective lightweight DLGSANet to solve image super-resolution. The proposed DLGSANet is mainly composed of several residual hybrid dynamic-Transformer groups (RHDTGs), where each RHDTG takes the hybrid dynamic-Transformer block (HDTB) as the basic module. The HDTB includes a simple yet effective multi-head dynamic local self-attention module (MHDLSA) for local feature extraction and a sparse global self-attention (SparseGSA) module for global feature extraction. In contrast to existing Transformers, the proposed HDTB not only extracts local features efficiently but also aggregates the most useful global features by a sparse global self-attention estimation method. By training the proposed DLGSANet in an end-to-end manner, we show that it has fewer network parameters and lower computational costs while achieving competitive performance against state-of-the-art ones on benchmarks in terms of accuracy.

**Acknowledgements.** This work has been partly supported by the National Key R&D Program of China (No. 2018AAA0102002), the National Natural Science Foundation of China (Nos. U22B2049, 62272233, 61925204), and the Fundamental Research Funds for the Central Universities (Nos. 30922010910, 30920041109).

## References

- [1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *ECCV*, 2018. 2
- [2] Pablo Arbeláez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011. 5
- [3] 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. 5
- [4] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *CVPR*, 2021. 1, 2
- [5] Xiaojie Chu, Liangyu Chen, , Chengpeng Chen, and Xin Lu. Improving image restoration by revisiting global information aggregation. In *ECCV*, 2022. 5
- [6] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019. 2, 5, 6
- [7] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 1, 2
- [8] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 1, 2
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1
- [10] Qi Han, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. On the connection between local attention and dynamic depth-wise convolution. In *ICLR*, 2022. 3
- [11] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng. Ode-inspired network design for single image super-resolution. In *CVPR*, 2019. 1
- [12] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 5, 8
- [13] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *ACM MM*, 2019. 2, 6, 7, 9
- [14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2
- [15] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1, 2
- [16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, 2016. 2
- [17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5
- [18] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2
- [19] Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. LAPAR: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. In *NeurIPS*, 2020. 2

- [20] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *ICCV Workshops*, 2021. 2, 3, 5, 6, 7, 9
- [21] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, 2017. 1, 2, 5, 6, 7, 9
- [22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2, 3
- [23] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 3
- [24] Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *ECCV*, 2020. 2, 6, 7, 9
- [25] Yusuke Matsui, Kota Ito, Yuji Aramaki, Toshihiko Yamasaki, and Kiyooharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *arXiv preprint arXiv:1510.04389*, 2015. 5
- [26] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *CVPR*, 2021. 1, 2, 5, 6, 9
- [27] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *ECCV*, 2020. 1, 2, 5, 6
- [28] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 3
- [29] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *CVPR*, 2017. 2
- [30] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 2, 3, 4
- [31] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, 2012. 5
- [32] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *ECCV*, 2022. 2, 3, 5, 6, 7, 9
- [33] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 1, 2, 5, 6, 9
- [34] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 5