This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Progressive Focused Transformer for Single Image Super-Resolution

Wei Long[1], Xingyu Zhou[1], Leheng Zhang[1], Shuhang Gu[1*]
[1]University of Electronic Science and Technology of China
weilong@std.uestc.edu.cn, shuhanggu@gmail.com

https://github.com/LabShuHangGU/PFT-SR

Abstract

Transformer-based methods have achieved remarkable results in image super-resolution tasks because they can capture non-local dependencies in low-quality input images. However, this feature-intensive modeling approach is computationally expensive because it calculates the similarities between numerous features that are irrelevant to the query features when obtaining attention weights. These unnecessary similarity calculations not only degrade the reconstruction performance but also introduce significant computational overhead. How to accurately identify the features that are important to the current query features and avoid similarity calculations between irrelevant features remains an urgent problem. To address this issue, we propose a novel and effective ***Progressive Focused Transformer (PFT)*** *that links all isolated attention maps in the network through* ***Progressive Focused Attention (PFA)*** *to focus attention on the most important tokens. PFA not only enables the network to capture more critical similar features, but also significantly reduces the computational cost of the overall network by filtering out irrelevant features before calculating similarities. Extensive experiments demonstrate the effectiveness of the proposed method, achieving state-of-the-art performance on various single image super-resolution benchmarks.*

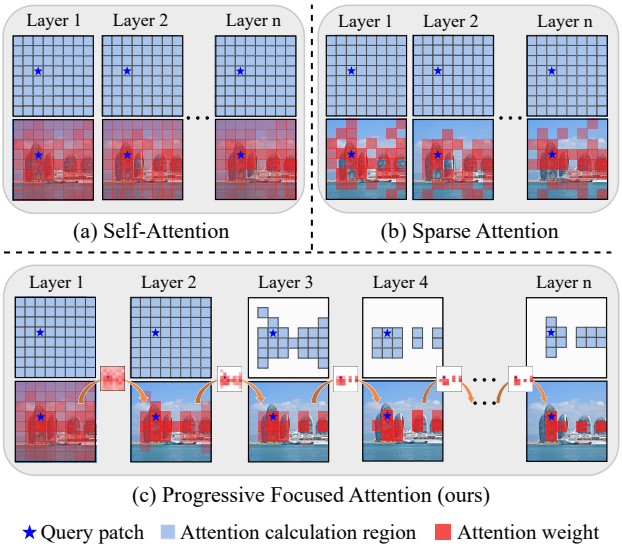


Figure 1. Comparison of different attention mechanisms. (a) Vanilla self-attention calculates attention weights in the whole window and generates non-zero weights for both highly relevant and less relevant tokens. (b) Sparse Attention is able to filter out the impact of less relevant tokens with small weights, but still requires the calculation of attention weights for all tokens. (c) Progressive Focused Attention connects isolated attention maps, leveraging attention weights to skip unnecessary computations and better aggregate relevant tokens.

1. Introduction

Image Super-Resolution (SR) aims to reconstruct high-resolution details and textures from Low-Resolution (LR) images. Effective SR techniques not only enhance visual quality but also enable more meaningful information extraction from high-resolution outputs. It has critical applications in fields such as medical imaging, satellite remote sensing, and video surveillance.

Several classical Convolutional Neural Networks (CNNs) [10, 11, 15, 23] designed for SR have achieved promising reconstruction results. Due to the limitations of convolution operations, these methods can capture only local feature information. Many studies have focused on enabling networks to achieve broader feature interaction to improve image restoration. Recently, Transformer-based methods [6, 22, 42, 46] have been increasingly applied to image reconstruction, owing to their ability to effectively leverage long-range dependencies, thereby enhancing the recovery of detailed information in images.

Transformer-based super-resolution methods utilize cascaded self-attention blocks to recover the missing High-Resolution (HR) details. At the core of the self-attention

*Corresponding author

block is the attention map, which determines how the input features are aggregated to generate output features. Due to the quadratic computational complexity of the self-attention mechanism, most of the existing methods confine attention computation within local windows and only aggregate a limited number of tokens to update image features. Therefore, how to exploit information of relevant tokens while alleviating the disruption from irrelevant tokens is critical in Transformer-based SR models. For the pursuit of high-quality SR results, one category of research tries to enrich the information source by enlarging the window size [6, 46], leveraging cross-window information [7], and introducing an external token dictionary [41]. Since more information can be incorporated into the weighted feature updating process, these methods have shown promising capability in delivering superior SR results. However, calculating similarities between more tokens inevitably leads to extra computational footprint. These methods still need to compromise on model overhead and introduce limited extra information. Another category of studies investigates sophisticated weighting strategies to make better use of input information [5, 28]. Instead of combining all the tokens according to the normalized productive similarities, as in a vanilla self-attention block, various elaborately designed weighting mechanisms have been exploited to make better use of highly relevant tokens while at the same time filtering out less relevant tokens. While the overall concept of seeking more rational weights is highly appealing, the challenge of obtaining enhanced attention weights remains an open problem in the literature.

In this paper, we link the attention maps between adjacent layers and propose a Progressive Focused Attention (PFA) block to better leverage input tokens across a wider range. Instead of combining input features merely according to the calculated attention map $\mathbf{A}^l_{cal}$, we inherit attention information from the previous block in a productive manner $\mathbf{A}^l = Norm\left(\mathbf{A}^{l-1} \odot \mathbf{A}^l_{cal}\right)$, where $Norm(\cdot)$ is the row-wise normalization operation. Such an attention inheritance mechanism enables our method to make a comprehensive measurement of token similarity: tokens that are consistently similar to each other are empowered with larger weights to help each other while the weights of irrelevant tokens are suppressed during the cascaded multiplications. Furthermore, in addition to boosting weights for important tokens and suppressing weights for less relevant tokens, another advantage of our PFA lies in its calculation pre-filtering capability. Specifically, since the final attention map $\mathbf{A}^l$ will be multiplied by the previous attention map $\mathbf{A}^{l-1}$, small weights in $\mathbf{A}^{l-1}$ are a sign of unimportant tokens, for which the weight calculation process can be skipped. As shown in Fig.1, the previous attention map not only helps our PFA block to generate better weights which emphasize highly relevant tokens, but also enables it to skip unnecessary similarity calculations so that we are able to afford the computational overhead of large window. Therefore, with comparable computational resources, our SR model could employ a larger window size to leverage input tokens from a broader range.

The main contributions of this paper are as follows:

- We propose the idea of Progressive Focused Attention (PFA) which inherits attention weights from previous layers in a productive manner. PFA not only avoids unnecessary similarity computation with less relevant tokens but also endows highly relevant tokens with larger weights to better enhance the input feature.
- We instantiate a Progressive Focused Transformer (PFT) for image super-resolution, which takes the progressive focusing benefits of PFA and systematically arranges computational resources for different layers. This property enables the PFT to leverage the merit of large window while maintaining model efficiency.
- We conduct extensive experiments on super-resolution benchmarks. Our superior experimental results over recent state-of-the-art methods as well as our detailed ablation studies demonstrate the effectiveness of our method.

2. Related Work

Transformer-based SR. Transformer models were initially used for natural language processing [9, 33]. Subsequently, they have attracted extensive research interest in the field of computer vision [3, 13, 14, 24, 36]. Due to its excellent feature extraction capabilities, Transformers have also been introduced to low-level visual tasks, including single image super-resolution [21, 37, 43]. Classical SR models [18, 19, 30, 40] based on CNN architectures were limited by narrow receptive fields and limited feature interaction capabilities, and their effectiveness was gradually surpassed by Transformer-based approaches. For example, IPT [4] implements a pre-trained Transformer-based model for better super-resolution. SwinIR [22] improves image quality by using shifted window attention and residual blocks, effectively capturing a broad range of features for restoration. Subsequently, CAT [7] improves image restoration by using rectangle-window self-attention and axial shifting to enhance cross-window interaction. To address the perceptual limitations of window-based methods, several approaches have been proposed. HAT [6] combines channel and window-based attention to overcome these limitations and enhance cross-window interactions for superior image reconstruction. Another approach is ATD [41], which addresses the limitations of local windows by using an Adaptive-Token Dictionary and category-based self-attention to incorporate global information and enhance features. In addition, IPG [31] improves SR by prioritizing detail-rich pixels and using flexible local and global graphs to enhance reconstruction. Unlike existing methods, we

propose a PFT method that can better focus on key tokens and filter out irrelevant features through PFA, thereby reducing the impact of irrelevant features on SR results and the computational effort.

Transformer with Sparse Attention. The dense attention maps in Transformers may introduce features from irrelevant locations, which interferes with the effectiveness of feature extraction. To address this, many approaches explore how to leverage sparse attention calculations to better utilize relevant information. For example, NLSA [28] combines non-local operations with sparse representations, using dynamic sparse attention to focus on pertinent features, thereby improving efficiency and robustness while reducing computational costs. ART [39] enhances image restoration by combining both dense and sparse attention mechanisms, expanding the receptive field to enable more effective interactions and better feature representation. Additionally, a class of methods [35, 45] explicitly selects the most relevant positions in the attention map for retention, filtering out irrelevant information that might interfere with feature aggregation. Building on this, DRSformer [5] further develops the idea of sparse selection by introducing a learnable top-k selection operator, that adaptively retains the most relevant self-attention values, thus improving image restoration. We propose the PFT method, which connects isolated attention maps across the entire network to efficiently identify the positions each attention map collectively focuses on. PFA performs a Hadamard product on adjacent attention maps to pinpoint the most relevant attention locations, thereby enhancing the filtering of irrelevant information.

3. Progressive Focused Transformer

3.1. Preliminaries

Self-Attention. The self-attention mechanism [33] is a fundamental part of Transformer, which computes the similarity between tokens and aggregates features based on the corresponding attention weights.

Given the query matrix $\mathbf{Q} \in \mathbb{R}^{N\times d}$, key matrix $\mathbf{K} \in \mathbb{R}^{N\times d}$, the attention weights are calculated as follows:

$$\mathbf{A}_{sa} = \text{Softmax}\left(\mathbf{Q}\mathbf{K}^\top/\sqrt{d}\right), \tag{1}$$

where the i-th row in $\mathbf{A}_{sa}$ represents normalized weights to aggregate value tokens $\mathbf{V} \in \mathbb{R}^{N\times d}$ and the output of self-attention module is obtained by $\mathbf{O}_{SA} = \mathbf{A}_{sa}\mathbf{K}$. Although the exponential function in Softmax could suppress irrelevant tokens to some extent, the above vanilla self-attention still apply dense weights which utilize all the input tokens in the window to generate the final output. Moreover, another feature of the above self-attention lies in its computational overhead. The complexity of Eq. (1) grows quadratically with the number of tokens N. Therefore, most existing methods need to perform self-attention calculations in small windows to manage this computational burden.

Sparse Attention. To mitigate the negative impact of irrelevant features in the dense attention map, numerous sparse attention methods have been proposed. Generally speaking, the overall idea of sparse attention is to introduce a selection operation to sparsify the attention weights:

$$\mathbf{A}_{ssa} = \text{Softmax}\left(\mathcal{S}\left(\mathbf{Q}\mathbf{K}^\top/\sqrt{d}\right)\right), \tag{2}$$

where $\mathcal{S}(\cdot)$ is a sparsification operation which removes the weights for less relevant tokens. The most commonly adopted sparsify operation is the top-k selection, which only allows K non-zero values in each row. Despite its strong capability of irrelevant token suppression, top-k selection only adjusts aggregation weights according to a single step of similarity calculation and has limited ability to make effective use of highly relevant tokens.

3.2. Progressive Focused Attention

As reviewed in the previous section, although sparse attention methods are able to improve the vanilla self-attention by alleviating the negative impact of irrelevant tokens, they still have limited capability in further distinguishing highly relevant tokens from mildly relevant tokens. Moreover, both the vanilla and sparse attention methods fail to identify the relevant feature regions prior to similarity calculation, resulting in a significant amount of unnecessary calculations, which impedes the incorporation of additional input tokens.

In this paper, we propose Progressive Focused Attention, which links the attention maps between adjacent layers to better aggregate relevant features. Generally, the proposed PFA advances the existing methods in two aspects. Firstly, PFA makes use of attention maps from previous layers to better weight input features. The weights of highly relevant tokens that are consistently similar to each other across layers will be enhanced during the cascade multiplication process. Secondly, PFA leverages the attention map of previous layers to identify irrelevant features for saving computational overhead. This selective calculation mechanism enables us to conduct self-attention within a larger window. In the remaining part of this section, we first introduce the basic idea of our PFA, which inherits the attention map from the previous layers through a Hadamard product. Then, we introduce how we could further leverage previous attention maps to reduce computational overhead.

Progressive Attention Across Layers. As introduced in Eq. (1) and (2), the previous methods obtain attention weights merely according to the calculated similarities between tokens from the current layer. The standard similarity calculation has limited discriminative capability in boosting highly relevant tokens while suppressing less relevant tokens. To address this issue, we propose to inherit attention

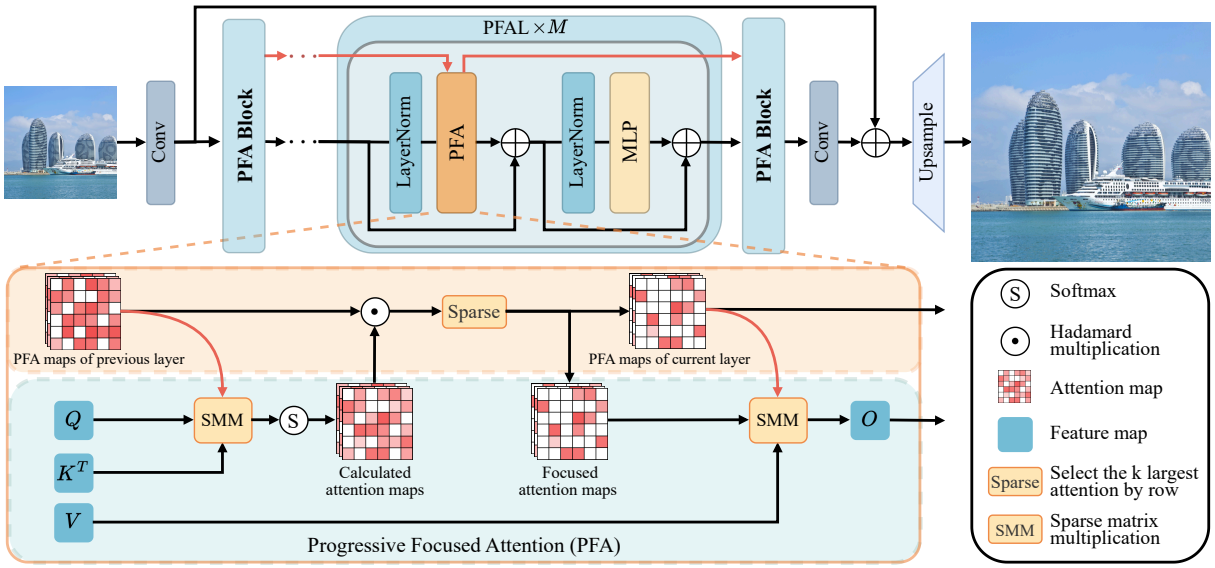


Figure 2. The overall architecture of PFT. PFA Block consists of M Progressive Focused Attention Layers (PFAL). Each PFA takes both image features and aggregated PFA maps from previous layers as input. Sparse Matrix Multiplication (SMM) ensures each row of Q interacts only with sparse columns of K^T, producing calculated attention maps. After applying the Hadamard product and sparse focusing, PFA maps of the current layer are obtained and used with the V matrix in an SMM operation to generate attention-aggregated features.

weights from previous layers in a productive manner:

$$\mathbf{A}^l = Norm(\mathbf{A}^{l-1} \odot \mathbf{A}^l_{cal}), \tag{3}$$

where $\mathbf{A}^l_{cal}$ is the calculated attention map for the l-th layer according to the vanilla self-attention mechanism as in Eq. (1); $\mathbf{A}^{l-1}$ is the attention map of the previous layer; $Norm(\cdot)$ represents the row-wise normalization operation, which ensures the summation of each row to be 1. As indicated in Eq. (3), the final attention map in the l-th layer is obtained by multiplying the calculated similarities with the attention map from the previous layer. Such a mechanism enables us to make a comprehensive assessment of token similarity: the weights for less relevant tokens will be wiped out during the successive multiplication process, while important tokens with high similarity values across layers will be assigned with larger weights after normalization. As will be validated in our ablation studies, with improved feature aggregation weights, such a Progressive Attention mechanism could deliver better SR results over the top-k sparse attention as well as the standard self-attention block.

Progressive Focused Attention. Since all the normalized similarities are less than 1, the above strategy of multiplicative attention inheritance could gradually filter out the impact of irrelevant tokens and obtain sparse weights as the network depth increases. Actually, based on the previous attention map $\mathbf{A}^{l-1}$, we are able to identify unimportant positions before calculation. Therefore, we can avoid unnecessary computations with a Sparse Matrix Multiplication (SMM) operation:

$$\mathbf{A}^l_{sc} = Softmax(\mathbf{\Psi}(\mathbf{Q}^l, (\mathbf{K}^l)^\top, \mathbf{I}^{l-1})), \tag{4}$$

where $\mathbf{\Psi}$ denotes the SMM operation; $\mathbf{I}^{l-1} \in \mathbb{R}^{N\times N}$ is a sparse index matrix; we calculate $\mathbf{A}^l_{sc}(i,j)$ with $\mathbf{Q}^l(i,:)$ and $\mathbf{K}^l(j,:)^T$ only if $\mathbf{I}^{l-1}(i,j) = 1$, otherwise we set $\mathbf{A}^l_{sc}(i,j) = 0$. Note that after the SMM operation we still need to normalize each row of the calculation results using the Softmax function. For an all-ones indexing matrix, the SMM operation in Eq. (4) is equivalent to the standard attention calculation process as in Eq. (1); However, if we are able to obtain a sparse $\mathbf{I}^{l-1}$ prior to the calculation process, a significant portion of computations can be skipped. Due to our progressive attention mechanism, we utilize $\mathbf{A}^{l-1}$ to indicate skippable positions. Specifically, inspired by the success of top-k sparse attention, we only keep K^l non-zero attention weights for each layer and generate the indexing matrix $\mathbf{I}^l$ according to the non-zero positions in $\mathbf{A}^l$. To sum up, the attention map of our progressive focused attention is achieved through the following formula:

$$\begin{cases} \mathbf{A}^l_{sc} &= Softmax(\mathbf{\Psi}(\mathbf{Q}^l, (\mathbf{K}^l)^\top, \mathbf{I}^{l-1})), \\ \mathbf{A}^l &= \mathcal{S}_{K^l}(Norm(\mathbf{A}^l_{sc} \odot \mathbf{A}^{l-1})), \\ \mathbf{I}^l &= \mathrm{Sign}(\mathbf{A}^l), \end{cases} \tag{5}$$

Where $\mathcal{S}_{K^l}(\cdot)$ is the sparsification operation that selects the top K^l values in each row and $\mathrm{Sign}(\cdot)$ generates a binary index matrix according to the sign of input matrix. Then,

we can also use the SMM operation to calculate the output matrix of the l-th layer:

$$\mathbf{O}^l = \mathbf{\Psi}(\mathbf{A}^l, \mathbf{V}^l, \mathbf{I}^l). \tag{6}$$

For the first attention block, in which we do not have previous attention map, we simply set $\mathbf{A}^0$ and $\mathbf{I}^0$ as all-ones matrix and calculate $\mathbf{A}^1$ as standard self-attention in Eq. (1) to aggregate input features. For subsequent layers, we exploit the attention map from previous layer to generate better weights and filter out unnecessary computations as in Eq. (5). Since tokens which have been determined as irrelevant tokens to a specific query token in previous layers will not be considered in the following layers at all, to avoid filtering-out important tokens at early stages, we set $K^1 = N$ and gradually reduce the nonzero values with a focus ratio: $K^l = \alpha K^{l-1}$, where K represents the number of retained attention values within the window. With $0 < \alpha < 1$, our cascade PFA layers gradually focus on important candidates and reduce the number of key-tokens to be considered for each input query-token, resulting in significant reduction in computational burden for each window. Notably, using standard matrix multiplication to implement the SMM operation is highly inefficient. Therefore, we developed efficient CUDA kernels for SMM. This implementation significantly enhances the efficiency of our model, enabling PFT to achieve inference speeds comparable to those of existing methods.

3.3. The Overall Network Architecture

As illustrated in Fig. 2, the overall architecture of PFT follows the structure adopted in recent state-of-the-art Transformer-based SR models [6, 22, 41]. While, we replace the standard self-attention blocks with our proposed PFA block for the pursuit of better SR results. Notably, we also incorporate the shifted window approach from the Swin Transformer, though it is omitted in the illustration for simplicity. Thus, the attention map in PFA is passed from the first layer to the last through two pathways, divided by the shift operation into even and odd layers for focused processing. Additionally, we incorporate LePE positional encoding [12] during attention computation. For classical SR, the PFT network consists of 6 blocks, with the number of attention layers in each block being [4, 4, 4, 6, 6, 6]. The model utilizes multi-head attention with 6 attention heads, and the total number of channels is 240. The window size is 32×32, and the number of retained attention values in each block is [1024, 256, 128, 64, 32, 16], respectively. For lightweight SR, the PFT-light network is composed of five blocks, with the number of attention layers in each block being [2, 4, 6, 6, 6]. It employs 4 attention heads, with a total of 52 channels. The window size is 32×32, and the number of retained attention values in each block follows the pattern [1024, 256, 128, 64, 32].

Analysis of Computational Complexity. Self-Attention and sparse attention have the same computational cost, as both measure similarity with neighboring positions by interacting with tokens within non-overlapping windows. When the input feature map size is $h \times w \times C$ and the window size is $W \times W$, the feature map is divided into $\frac{h}{W} \times \frac{w}{W}$ windows, each with $W \times W$ tokens. Assuming that the network has a total of L attention layers, the computational complexity of the attention component for the entire network is:

$$\Omega(SA) = 4hwLC^2 + 2W^2hwLC. \tag{7}$$

PFA can control the proportion of key tokens that are most relevant to the query features by adjusting the retention ratio α. The index matrix $\mathbf{I}$ allows shallow attention aggregation information to be carried into deeper layers for further focusing, resulting in a gradually decreasing K. For a network with L attention layers, we assume that the number of retained attention values within the window decreases, as expressed by the following formula: $K^l = \alpha K^{l-1}$, where $0 < \alpha < 1$ and K^1 is generally set to W^2. It can be represented as $K^l = W^2\alpha^{l-1}$. The overall computational complexity of the model is:

$$\Omega(PFA) = \sum_{l=1}^{L} \left(4hwC^2 + 2\alpha^{(l-1)}W^2hwC\right), \tag{8}$$

where $\alpha^{(l-1)}$ controls the decay of the focus ratio across layers, with $0 < \alpha < 1$. As l increases, the focus ratio progressively decreases. The first term, $4hwC^2$, represents the computational complexity of the projection layers. The second term, $2\alpha^{(l-1)}W^2hwC$, accounts for the complexity of computing the similarity between the query and the selected key tokens, as well as redistributing the corresponding value information at the focused positions. Obviously, since α is less than 1, the complexity of this term decreases exponentially with depth. For example, with $\alpha = 0.5$, the computational complexity is reduced to just 6.25% of the original after four decay steps.

4. Experiments

4.1. Experiment Setting

We use the DF2K dataset, which combines DIV2K [32] and Flickr2K [23], as our training set. To ensure fair comparisons, we adopt the same training configuration as utilized in recent SR studies [6, 31, 41]. Our model is trained for 500K iterations using the AdamW optimizer and an initial learning rate of 2×10^{-4}. The input patch size for training is fixed at 64×64, and we employ a MultistepLR scheduler, which reduces the learning rate by half at specified iterations $[250000, 400000, 450000, 475000]$. The batch size is set to 32. We evaluate our method on five standard benchmark datasets: Set5 [2], Set14 [38], BSD100 [26],

Method	Scale	Params	FLOPs	Set5		Set14		BSD100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
EDSR [23]	×2	42.6M	22.14T	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
RCAN [44]	×2	15.4M	7.02T	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
HAN [29]	×2	63.6M	7.24T	38.27	0.9614	34.16	0.9217	32.41	0.9027	33.35	0.9385	39.46	0.9785
IPT [4]	×2	115M	7.38T	38.37	-	34.43	-	32.48	-	33.76	-	-	-
SwinIR [22]	×2	11.8M	3.04T	38.42	0.9623	34.46	0.9250	32.53	0.9041	33.81	0.9433	39.92	0.9797
CAT-A [7]	×2	16.5M	5.08T	38.51	0.9626	34.78	0.9265	32.59	0.9047	34.26	0.9440	40.10	0.9805
ART [39]	×2	16.4M	7.04T	38.56	0.9629	34.59	0.9267	32.58	0.9048	34.30	0.9452	40.24	0.9808
HAT [6]	×2	20.6M	5.81T	38.63	0.9630	34.86	0.9274	32.62	0.9053	34.45	0.9466	40.26	0.9809
IPG [31]	×2	18.1M	5.35T	38.61	0.9632	34.73	0.9270	32.60	0.9052	34.48	0.9464	40.24	0.9810
ATD [41]	×2	20.1M	6.07T	38.61	0.9629	34.95	0.9276	32.65	0.9056	34.70	0.9476	40.37	0.9810
PFT (Ours)	×2	19.6M	5.03T	38.68	0.9635	35.00	0.9280	32.67	0.9058	34.90	0.9490	40.49	0.9815
EDSR [23]	×3	43.0M	9.82T	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17	0.9476
RCAN [44]	×3	15.6M	3.12T	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
HAN [29]	×3	64.2M	3.21T	34.75	0.9299	30.67	0.8483	29.32	0.8110	29.10	0.8705	34.48	0.9500
IPT [4]	×3	116M	3.28T	34.81	-	30.85	-	29.38	-	29.49	-	-	-
SwinIR [22]	×3	11.9M	1.35T	34.97	0.9318	30.93	0.8534	29.46	0.8145	29.75	0.8826	35.12	0.9537
CAT-A [7]	×3	16.6M	2.26T	35.06	0.9326	31.04	0.8538	29.52	0.8160	30.12	0.8862	35.38	0.9546
ART [39]	×3	16.6M	3.12T	35.07	0.9325	31.02	0.8541	29.51	0.8159	30.10	0.8871	35.39	0.9548
HAT [6]	×3	20.8M	2.58T	35.07	0.9329	31.08	0.8555	29.54	0.8167	30.23	0.8896	35.53	0.9552
IPG [31]	×3	18.3M	2.39T	35.10	0.9332	31.10	0.8554	29.53	0.8168	30.36	0.8901	35.53	0.9554
ATD [41]	×3	20.3M	2.69T	35.11	0.9330	31.13	0.8556	29.57	0.8176	30.46	0.8917	35.63	0.9558
PFT (Ours)	×3	19.8M	2.23T	35.15	0.9333	31.16	0.8561	29.58	0.8178	30.56	0.8931	35.67	0.9560
EDSR [23]	×4	43.0M	5.54T	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148
RCAN [44]	×4	15.6M	1.76T	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
HAN [29]	×4	64.2M	1.81T	32.64	0.9002	28.90	0.7890	27.80	0.7442	26.85	0.8094	31.42	0.9177
IPT [4]	×4	116M	1.85T	32.64	-	29.01	-	27.82	-	27.26	-	-	-
SwinIR [22]	×4	11.9M	0.76T	32.92	0.9044	29.09	0.7950	27.92	0.7489	27.45	0.8254	32.03	0.9260
CAT-A [7]	×4	16.6M	1.27T	33.08	0.9052	29.18	0.7960	27.99	0.7510	27.89	0.8339	32.39	0.9285
ART [39]	×4	16.6M	1.76T	33.04	0.9051	29.16	0.7958	27.97	0.7510	27.77	0.8321	32.31	0.9283
HAT [6]	×4	20.8M	1.45T	33.04	0.9056	29.23	0.7973	28.00	0.7517	27.97	0.8368	32.48	0.9292
IPG [31]	×4	17.0M	1.30T	33.15	0.9062	29.24	0.7973	27.99	0.7519	28.13	0.8392	32.53	0.9300
ATD [41]	×4	20.3M	1.52T	33.10	0.9058	29.24	0.7974	28.01	0.7526	28.17	0.8404	32.62	0.9306
PFT (Ours)	×4	19.8M	1.26T	33.15	0.9065	29.29	0.7978	28.02	0.7527	28.20	0.8412	32.63	0.9306

Table 1. Quantitative comparison (PSNR/SSIM) with state-of-the-art methods on **classical SR** task. The best and second best results are colored with red and blue.

Urban100 [16], and Manga109 [27]. In addition, the computational costs of all models in this paper are measured at an output resolution of 1280 × 640.

4.2. Comparison with State-of-the-Art Methods

We compare the performance of our model with various SR baselines on widely used benchmark datasets, including Set5 [2], Set14 [38], BSD100 [26], Urban100 [16], and Manga109 [27]. These SR methods include both classical and state-of-the-art approaches, such as EDSR [23], RCAN [44], HAN [29], IPT [4], SwinIR [8], CAT [7], ART [39], HAT [6], IPG [31], and ATD [41]. We use PSNR and SSIM metrics to assess the performance of SR models at ×2, ×3, and ×4 upscaling factors. All computational costs are calculated at the output resolution of 1280×640, and PFT uses a window size of 32×32.

The results, shown in Tab. 1, indicate that the proposed PFT model achieves superior performance over ATD, with significantly reduced computational costs. Notably, PFT reduces computational load by **17.1%** compared to ATD, while attaining the highest PSNR in most datasets. For lightweight SR tasks, we further compare our model with other efficient SR methods, including CARN [1], IMDN [17], LAPAR [20], LatticeNet [25], SwinIR [8], SwinIR-NG [8], ELAN [42], and OmniSR [34]. As shown in Tab. 2, the proposed PFT-light outperforms the recently introduced ATD-light [41] on nearly all benchmark datasets, while reducing computational complexity by **20.1%**. On the ×2 Urban100 benchmark, PFT-light outperforms ATD-light by **0.40dB** and IPG-Tiny by **0.63dB**. For the ×4 SR experiments, PFT-light also outperforms IPG Tiny by **0.42dB** and ATD-light by **0.23dB** on the Urban100 test set. The superior performance of PFT can be attributed to its use of Progressive Focused Attention, which connects otherwise isolated attention maps across the network. This enables the model to focus on essential features, filter out irrelevant patches, substantially reduce computational costs, and support the use of larger window sizes.

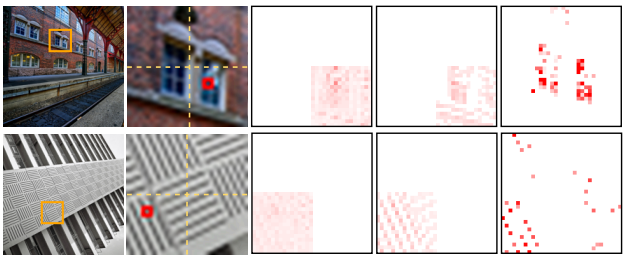


Figure 3. Visual comparison of attention distributions in the 18th layer. SA and top-k Attention distribute attention broadly, failing to focus on the most relevant areas. In contrast, PFA filters out irrelevant tokens and concentrates attention on key regions. By reducing computational costs, it enables the use of a larger 32×32 window for more extensive feature interactions.

Method	Scale	Params	FLOPs	Set5		Set14		BSD100		Urban100		Manga109	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CARN [1]	×2	1,592K	222.8G	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
IMDN [17]	×2	694K	158.8G	38.00	0.9605	33.63	0.9177	32.19	0.8996	32.17	0.9283	38.88	0.9774
LAPAR-A [20]	×2	548K	171G	38.01	0.9605	33.62	0.9183	32.19	0.8999	32.10	0.9283	38.67	0.9772
LatticeNet [25]	×2	756K	169.5G	38.15	0.9610	33.78	0.9193	32.25	0.9005	32.43	0.9302	-	-
SwinIR-light [22]	×2	910K	244G	38.14	0.9611	33.86	0.9206	32.31	0.9012	32.76	0.9340	39.12	0.9783
ELAN [42]	×2	582K	203G	38.17	0.9611	33.94	0.9207	32.30	0.9012	32.76	0.9340	39.11	0.9782
SwinIR-NG [8]	×2	1181K	274.1G	38.17	0.9612	33.94	0.9205	32.31	0.9013	32.78	0.9340	39.20	0.9781
OmniSR [34]	×2	772K	194.5G	38.22	0.9613	33.98	0.9210	32.36	0.9020	33.05	0.9363	39.28	0.9784
IPG-Tiny [31]	×2	872K	245.2G	38.27	0.9616	34.24	0.9236	32.35	0.9018	33.04	0.9359	39.31	0.9786
ATD-light [41]	×2	753K	348.6G	38.28	0.9616	34.11	0.9217	32.39	0.9023	33.27	0.9376	39.51	0.9789
PFT-light (Ours)	×2	776K	278.3G	38.36	0.9620	34.19	0.9232	32.43	0.9030	33.67	0.9411	39.55	0.9792
CARN [1]	×3	1,592K	118.8G	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
IMDN [17]	×3	703K	71.5G	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519	33.61	0.9445
LAPAR-A [20]	×3	544K	114G	34.36	0.9267	30.34	0.8421	29.11	0.8054	28.15	0.8523	33.51	0.9441
LatticeNet [25]	×3	765K	76.3G	34.53	0.9281	30.39	0.8424	29.15	0.8059	28.33	0.8538	-	-
SwinIR-light [22]	×3	918K	111G	34.62	0.9289	30.54	0.8463	29.20	0.8082	28.66	0.8624	33.98	0.9478
ELAN [42]	×3	590K	90.1G	34.61	0.9288	30.55	0.8463	29.21	0.8081	28.69	0.8624	34.00	0.9478
SwinIR-NG [8]	×3	1190K	114.1G	34.64	0.9293	30.58	0.8471	29.24	0.8090	28.75	0.8639	34.22	0.9488
OmniSR [34]	×3	780K	88.4G	34.70	0.9294	30.57	0.8469	29.28	0.8094	28.84	0.8656	34.22	0.9487
IPG-Tiny [31]	×2	878K	109.0G	34.64	0.9292	30.61	0.8470	29.26	0.8097	28.93	0.8666	34.30	0.9493
ATD-light [41]	×3	760K	154.7G	34.70	0.9300	30.68	0.8485	29.32	0.8109	29.16	0.8710	34.60	0.9505
PFT-light (ours)	×3	783K	123.5G	34.81	0.9305	30.75	0.8493	29.33	0.8116	29.43	0.8759	34.60	0.9510
CARN [1]	×4	1,592K	90.9G	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084
IMDN [17]	×4	715K	40.9G	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838	30.45	0.9075
LAPAR-A [20]	×4	659K	94G	32.15	0.8944	28.61	0.7818	27.61	0.7366	26.14	0.7871	30.42	0.9074
LatticeNet [25]	×4	777K	43.6G	32.30	0.8962	28.68	0.7830	27.62	0.7367	26.25	0.7873	-	-
SwinIR-light [22]	×4	930K	63.6G	32.44	0.8976	28.77	0.7858	27.69	0.7406	26.47	0.7980	30.92	0.9151
ELAN [42]	×4	582K	54.1G	32.43	0.8975	28.78	0.7858	27.69	0.7406	26.54	0.7982	30.92	0.9150
SwinIR-NG [8]	×4	1201K	63.0G	32.44	0.8980	28.83	0.7870	27.73	0.7418	26.61	0.8010	31.09	0.9161
OmniSR [34]	×4	792K	50.9G	32.49	0.8988	28.78	0.7859	27.71	0.7415	26.65	0.8018	31.02	0.9151
IPG-Tiny [31]	×4	887K	61.3G	32.51	0.8987	28.85	0.7873	27.73	0.7418	26.78	0.8050	31.22	0.9176
ATD-light [41]	×4	769K	87.1G	32.62	0.8997	28.87	0.7884	27.77	0.7439	26.97	0.8107	31.47	0.9198
PFT-light (Ours)	×4	792K	69.6G	32.63	0.9005	28.92	0.7891	27.79	0.7445	27.20	0.8171	31.51	0.9204

Table 2. Quantitative comparison (PSNR/SSIM) with state-of-the-art methods on **lightweight SR** task. The best and second best results are colored with red and blue.

Visual comparison of attention distribution. The attention distribution visualizations are shown in Fig. 3. The attention distributions of the Vanilla Self-Attention and top-k Attention methods are relatively scattered, failing to allocate attention to the regions most relevant to the current query. In contrast, our PFA method filters out irrelevant tokens and focuses on most important positions. It also lowers computation costs, enabling larger 32×32 windows for broader feature interactions.

Visual comparison of SR reconstruction results. To evaluate the quality of different super-resolution methods, we show visual examples in Fig. 4. These comparisons demonstrate the strength of our PFT approach in restoring sharp edges and fine textures from low-resolution images. For example, in img_001 and img_095, most methods struggle to reconstruct textures accurately, leading to noticeable distortions. Similarly, in the zebra and barbara images, many methods fail to recover details, resulting in blurry or incorrect textures. In contrast, our PFT effectively captures fine textures by focusing on the most relevant features. This ability to selectively attend helps restore clean edges and reduce artifacts, producing more accurate reconstructions.

4.3. Ablation Study

We conduct ablation studies on the proposed PFT-light model, with all models trained for 250k iterations on the DIV2K dataset at ×4 scale. Except for the experiments on window size, all other experiments use a window size of 16 × 16. In top-k Attention, the value of k is set to half the number of window elements, which is 128.

Effectiveness of Progressive Focused Attention. We conducted experiments to demonstrate the effectiveness of the proposed Progressive Attention and Progressive Focused Attention, with results presented in Tab. 3. These experiments compare the quantitative performance of ×4 super-resolution on the Set5, Urban100, and Manga109 datasets. Progressive Attention applies a straightforward Hadamard product connection of the attention maps. As shown, even the Progressive Attention provides notable performance improvements over Self-Attention and top-k Attention. Progressive Focused Attention achieved the best results with a **27.69%** reduction in computation, and notably increasing PSNR by **0.36dB** on the Urban100 dataset. The effectiveness of PFA lies in its ability to link adjacent attention maps across the network via the Hadamard product, allowing the model to reduce redundant computations and focus on the most relevant tokens.

Method	FLOPs	Set5	Urban100	Manga109
Vanilla Self-Attention	70.4G	32.28	26.26	30.62
Top-k Attention	70.4G	32.30	26.29	30.67
Progressive Attention	70.4G	32.35	26.41	30.78
Progressive Focused Attention	50.9G	32.41	26.62	30.85

Table 3. Ablation study on the proposed focused attention.

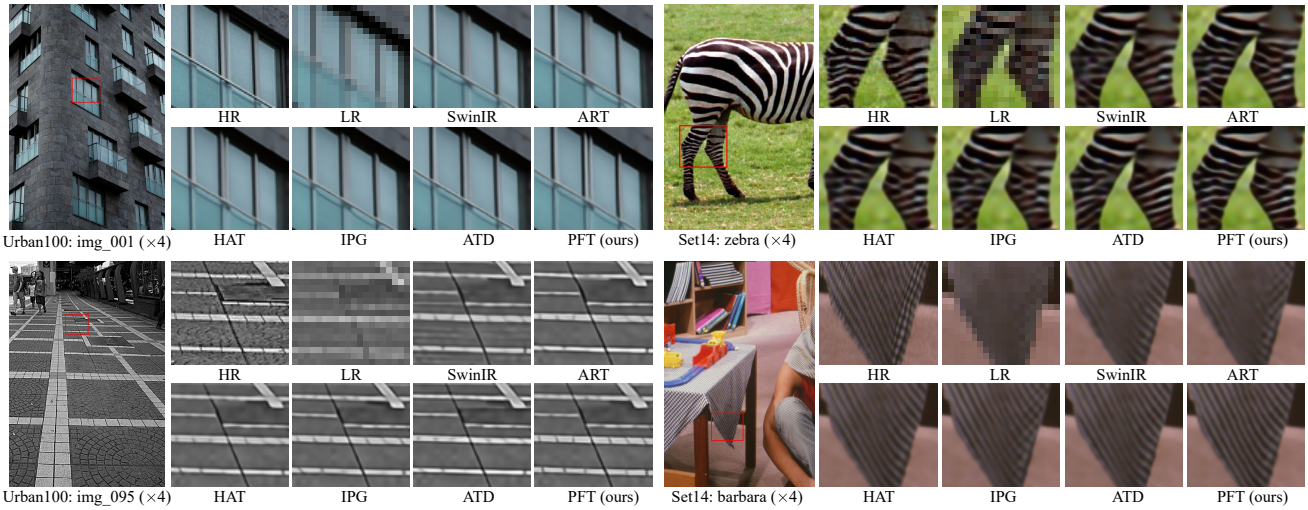


Figure 4. Visual comparison of SR reconstruction results.

Effects of focus ratio. As discussed in Sec. 3.2, the parameter α controls the focus degree of Progressive Focused Attention as network depth increases. A smaller α retains fewer attention positions, accelerating the focusing process. However, α is generally kept above 0.1, as a rapid reduction in attention retention may cause the network to make premature choices. As shown in Tab. 4, model performance is optimal when α is around 0.5. Experimental results indicates that both excessively large and small values of α can hinder effective attention focusing. A very small α may cause the network to overlook important tokens, leading to incomplete representations. On the other hand, a large α keeps too many irrelevant tokens, weakening the focus and limiting the network's ability to reduce noise.

Focus ratio	0.1	0.3	0.5	0.7	0.9
PSNR	26.48	26.58	26.62	26.61	26.56

Table 4. Ablation study on focus ratio for $\times 4$ SR, evaluated on Urban100.

Effects of different window sizes. The window size determines the model's scope of correlation extraction and significant impact on reconstruction performance. Previous studies have shown that larger windows yield better reconstruction results for image super-resolution tasks. However, due to the quadratic complexity of self-attention, previous methods find it challenging to use large windows effectively. Our PFA dynamically filters out patches irrelevant to the current query patch. This allows us to select the interaction scope before computing similarity. As a result, the overall computational load is greatly reduced. This enables us to use large windows while maintaining a low computational cost. As shown in Tab. 5, reconstruction performance improves as the window size increases. We ultimately select a window size of 32×32 to balance performance, model parameters, and computational cost.

Window size	Set5	Urban100	Manga109
8×8	32.33	26.40	30.71
16×16	32.41	26.62	30.85
32×32	32.49	26.81	30.93

Table 5. Ablation study on window sizes for $\times 4$ SR, evaluated on the Urban100 dataset, with PSNR as the evaluation metric.

5. Conclusion

We propose the PFT, which introduces a novel approach to linking attention maps across layers through the key innovation of the PFA mechanism, progressively refining the focus on relevant features. By inheriting and modifying attention maps from previous layers, the PFA mechanism reduces unnecessary similarity calculations, improving attention efficiency. This enables the model to prioritize the most relevant tokens while effectively suppressing the influence of irrelevant ones. Experiments show that PFT and PFT-light achieves leading performance in single-image super-resolution with lower computation. As a key component of the Progressive Focused Transformer, PFA has proven to be both effective and efficient for super-resolution tasks. We look forward to exploring its potential in other high-level vision tasks, as well as in natural language processing.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 62476051) and the Sichuan Natural Science Foundation (No. 2024NSFTD0041).

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European conference on computer vision*, pages 252–268, 2018. 6, 7

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 5, 6

[3] Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847*, 2021. 2

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021. 2, 6

[5] Xiang Chen, Hao Li, Mingqiang Li, and Jinshan Pan. Learning a sparse transformer network for effective image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5896–5905, 2023. 2, 3

[6] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22367–22377, 2023. 1, 2, 5, 6

[7] Zheng Chen, Yulun Zhang, Jinjin Gu, yongbing zhang, Linghe Kong, and Xin Yuan. Cross aggregation transformer for image restoration. In *Advances in Neural Information Processing Systems*, pages 25478–25490, 2022. 2, 6

[8] Haram Choi, Jeongmin Lee, and Jihoon Yang. N-gram in swin transformers for efficient lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2071–2081, 2023. 6, 7

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[10] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014. 1

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2015. 1

[12] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12124–12134, 2022. 5

[13] 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 *International Conference on Learning Representations*, 2020. 2

[14] Jinsheng Fang, Hanjiang Lin, Xinyu Chen, and Kun Zeng. A hybrid network of cnn and transformer for lightweight image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1103–1112, 2022. 2

[15] Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1823–1831, 2015. 1

[16] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 6

[17] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th acm international conference on multimedia*, pages 2024–2032, 2019. 6, 7

[18] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016. 2

[19] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016. 2

[20] 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. *Advances in Neural Information Processing Systems*, 33:20343–20355, 2020. 6, 7

[21] Yawei Li, Yuchen Fan, Xiaoyu Xiang, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. Efficient and explicit modelling of image hierarchies for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18278–18289, 2023. 2

[22] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1833–1844, 2021. 1, 2, 5, 6, 7

[23] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 136–144, 2017. 1, 5, 6

[24] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF con-*

ference on computer vision and pattern recognition, pages 457–466, 2022. 2

[25] Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 272–289. Springer, 2020. 6, 7

[26] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, pages 416–423. IEEE, 2001. 5, 6

[27] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia tools and applications*, 76:21811–21838, 2017. 6

[28] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2021. 2, 3

[29] 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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 191–207. Springer, 2020. 6

[30] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017. 2

[31] Yuchuan Tian, Hanting Chen, Chao Xu, and Yunhe Wang. Image processing gnn: Breaking rigidity in super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24108–24117, 2024. 2, 5, 6, 7

[32] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei 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*, pages 114–125, 2017. 5

[33] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 2, 3

[34] Hang Wang, Xuanhong Chen, Bingbing Ni, Yutian Liu, and Jinfan Liu. Omni aggregation networks for lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22378–22387, 2023. 6, 7

[35] Pichao Wang, Xue Wang, Fan Wang, Ming Lin, Shuning Chang, Hao Li, and Rong Jin. Kvt: k-nn attention for boosting vision transformers. In *European Conference on Computer Vision*, pages 285–302. Springer, 2022. 3

[36] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5791–5800, 2020. 2

[37] Jinsu Yoo, Taehoon Kim, Sihaeng Lee, Seung Hwan Kim, Honglak Lee, and Tae Hyun Kim. Enriched cnn-transformer feature aggregation networks for super-resolution. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 4956–4965, 2023. 2

[38] Roman Zeyde, Michael Elad, and Matan 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*, pages 711–730. Springer, 2012. 5, 6

[39] Jiale Zhang, Yulun Zhang, Jinjin Gu, Yongbing Zhang, Linghe Kong, and Xin Yuan. Accurate image restoration with attention retractable transformer. In *ICLR*, 2022. 3, 6

[40] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3262–3271, 2018. 2

[41] Leheng Zhang, Yawei Li, Xingyu Zhou, Xiaorui Zhao, and Shuhang Gu. Transcending the limit of local window: Advanced super-resolution transformer with adaptive token dictionary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2024. 2, 5, 6, 7

[42] Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. In *European Conference on Computer Vision*, pages 649–667. Springer, 2022. 1, 6, 7

[43] Xiang Zhang, Yulun Zhang, and Fisher Yu. Hit-sr: Hierarchical transformer for efficient image super-resolution. *arXiv preprint arXiv:2407.05878*, 2024. 2

[44] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 6

[45] Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun. Explicit sparse transformer: Concentrated attention through explicit selection. *arXiv preprint arXiv:1912.11637*, 2019. 3

[46] Yupeng Zhou, Zhen Li, Chun-Le Guo, Song Bai, Ming-Ming Cheng, and Qibin Hou. Srformer: Permuted self-attention for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12780–12791, 2023. 1, 2