FSDFormer: Progressive Rain Removal Network Based on Fourier-Spatial Dual Transformer

Shuying Huang[1] , Jiaxuan Yang[1] , Yong Yang[1,2]* , Weiguo Wan[3]
[1]Tiangong University, Tianjin, China
[2]Cangzhou Institute of Tiangong University, Cangzhou, China
[3]Jiangxi University of Finance and Economics, Nanchang, China
shuyinghuang2010@126.com, yjx4149@163.com, greatyangy@126.com, wanweiguo@jxufe.edu.cn

Abstract

Most rain removal methods based on deep learning typically adopt a single-stage network architecture to remove the rain streaks in rainy images by increasing the depth of the network. The increase in network depth will increase the computational complexity of the model, and the lack of guidance for intermediate features will lead to inaccurate feature learning. To address this issue, we proposed a progressive rain removal network based on Fourier-spatial dual Transformer, called FSDFormer. The network consists of multiple rain removal stages, each with the same structure, which can utilize background prior features to guide the network to reconstruct rainless images with more texture information. Each stage consists of a prior extraction module (PEM), a prior attention fusion module (PAFM), and a U-Net including multiple Fourier-spatial dual Transformers (FSD-Transformers). Firstly, PEM is constructed to extract the background prior features from the input rainy image or the output of each stage. Then, a PAFM is designed to reconstruct accurate image background features by utilizing background prior features to guide the network. Finally, U-Net extracts and reconstructs features at different scales by constructing multiple FSD-Transformers to obtain rainless features at each stage. Extensive experimental results on synthetic and real datasets have shown that the proposed method outperforms some state-of-the-art (SOTA) rain removal methods in terms of visual quality and quantitative indicators. The source code is available at https://github.com/yangjiaxuan6250/FSDFormer.

1 Introduction

The presence of rain streaks or raindrops in images results in the loss of scene information and low contrast, which will have a significant impact on some high-level vision tasks, such as target detection [Redmon, 2016], video surveillance [Viola and Jones, 2004], and scene understanding [Zhao *et al.*, 2017]. In recent years, research on image rain removal tasks in the field of low-level vision has attracted increasing attention from researchers.

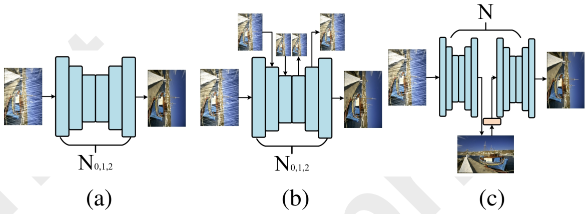


Figure 1: (a) single-stage architecture, (b) single-stage multi-input architecture, (c) our multi-stage architecture.

Early traditional methods [Kang *et al.*, 2011; Chen and Hsu, 2013; Li *et al.*, 2016] mostly constructed rain removal models by defining prior information of rain streaks or background image to obtain rainless images. Later, convolutional neural network (CNN) based methods [Zhang and Patel, 2018; Li *et al.*, 2018; Yang *et al.*, 2019] demonstrated excellent performance in learning complex mapping relationships between rainy and clean images due to their powerful feature representation capabilities, effectively handling rain streaks of different shapes, sizes, and densities. Recently, the Transformer method that can model non local information [Wang *et al.*, 2022] has further improved the performance of rain removal tasks.

Currently, most rain networks [Xiao *et al.*, 2022a; Chen *et al.*, 2023b] adopt two single-stage architectures, as shown in Figs.1 (a) and (b), which cannot utilize the explicit information present in multi-scale images. Figure 1 (a) does not use external information to guide intermediate features in the feature learning process. Figure 1 (b) can improve the accuracy of the mapping relationship by adding multiple inputs in the middle layers of the network. In addition, reference [Chen *et al.*, 2024a] improved the above structure by using a stacked U-Net structure to enhance the performance of the model, which resulted in a significant increase in computation and parameter count, but the performance improvement was not significant. Based on the above analysis, we constructed a multi-stage network architecture as shown in Figure 1 (c). Figure 2 shows a comparison of the performance of three architectures in rain removal tasks. Each architec-

*Corresponding author.

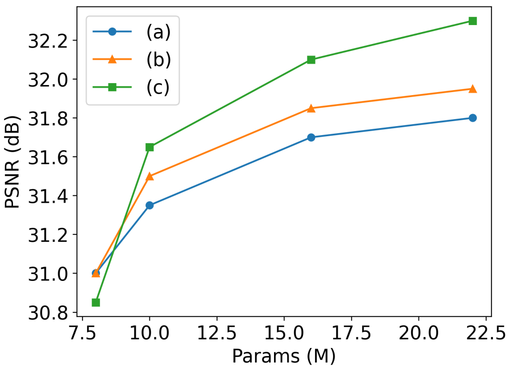


Figure 2: Performance comparison of three different architectures with increasing number of parameters. (a) single-stage architecture, (b) single-stage multi-input architecture and (c) multi-stage architecture.

ture uses the same number of feature extraction layers from Restormer [Xiao *et al.*, 2022a]. From the figure, it can be observed that as the number of parameters increases, the performance of our method improves faster. This also indicates that Figure 1(3) is effective. However, current methods mostly focus on feature extraction in the spatial domain, using CNN to extract local features and transformer structure to extract global features. The Transformer structure only uses a larger receptive field for feature extraction, which is still limited in global feature extraction. Therefore, to enhance the model's ability to extract global features, we introduce frequency domain feature extraction into the model.

Based on the above analysis, we propose a progressive rain removal network based on Fourier-spatial dual Transformer, named FSD-Former, which includes multiple rain removal stages with the same structure. In each stage, image background prior information is extracted and utilized to guide the learning of background features. Specifically, each rain removal stage consists of a PEM, a PAFM, and a U-Net that includes FSD-transformers. Firstly, a PEM is constructed to extract background prior features from the input rainy image or coarse rainless image from the previous stage. Then, a PAFM is designed, which utilizes prior background features to guide the network to learn background features with rich information. Next, a FSD-Transformer U-net is constructed to learn and reconstruct the different scale features in spatial and frequency domains by utilizing multiple FSD-Transformers, in order to output the rain removal results for each stage. The main contributions of this paper are as follows.

- A progressive rain removal network called FSD-Former is proposed, which gradually obtains clear rainless images by constructing multiple rain removal stages with the same structure.
- In each rain removal stage, PAFM is first constructed based on prior background features as guidance to achieve preliminary rain removal. Then, a U-net containing multiple FSD-Transformers is constructed to obtain refined background features to obtain the rain removal results for each stage.
- FSD-Transformer is designed to extract local features in the spatial domain and enhance global features in the frequency domain by constructing a local feature enhancement block (LFEB), a Fourier-spatial dual attention block (FDAB), and a Fourier enhancement gated block (FEGB).

2 Proposed Method

2.1 Overall Structure

In this section, a FSDFormer consisting of multiple rain removal stages with the same structure is proposed to achieve gradual removal of rain streaks, and the specific structure is shown in Figure 3. The structure of each stage is mainly composed of three modules: prior extraction module (PEM), prior attention fusion module (PAFM), and Fourier-spatial dual Transformer (FSD-Transformer).

In the first stage, the input rain image $I_R$is first fed in parallel to a 3×3 convolutional layer and a PEM [27] to obtain shallow features F_0 and prior background features F_p^1. This module is designed to extract a priori information from the rain map. Then, the two features are fused through the constructed PAFM to achieve the enhancement of background features. Next, a U-shaped network containing multiple FSD-Transformers is constructed to achieve fine-grained extraction and reconstruction of features at different scales. Finally, a 3×3 convolutional layer and a residual operation are used to obtained the initial rain removal result I_B^1. The specific operations can be expressed by the following equations.

$$F_0 = Conv_{3\times3}(I_R), F_P^1 = PEM(I_R), \tag{1}$$

$$F_u^i = Unet_i(PAFM(F_0, F_P^1)), i = 1, \tag{2}$$

$$I_B^1 = Conv_{3\times3}(F_u^1) + I_R, \tag{3}$$

where $Conv_{3\times3}(\cdot)$ represents a 3×3 convolution operation, $PEM(\cdot)$and $PAFM(\cdot)$ represent the operations of PEM and PAFM, and $Unet_i(\cdot)$ represents the U-shape network in the $i-th$ stage, and its output is F_u^i. To reduce the loss of features, the second and third stages uses the output of the U-shaped network in the previous stage and the initial shallow features as inputs to supplement the features and further refine the rain removal results. The operations of the subsequent two stages can be represented as follows.

$$F_p^i = PEM(I_b^{i-1}), i = 2, 3, \tag{4}$$

$$F_u^i = Unet_i(Concat(F_u^{i-1}, F_0), F_p^i), i = 2, ..., N, \tag{5}$$

$$I_B^i = Conv_{3\times3}(Concat(F_u^i, F_u^{i-1}, ..., F_u^1)) + I_R, \tag{6}$$

where $Concat(\cdot)$ represents the concatenation operation. I_B^i is the rain removal result from the $i-th$ stage, and $I_B^N$is the final rain removal result, denoted as I_B.

2.2 Prior Attention Fusion Module (PAFM)

To guide the network to learn rich background features, PAFM is designed to achieve the fusion of prior background features and shallow features by establishing a background prior attention matrix. The architecture of PAFM is illustrated in Figure 4. Firstly, the shallow features are sent to two

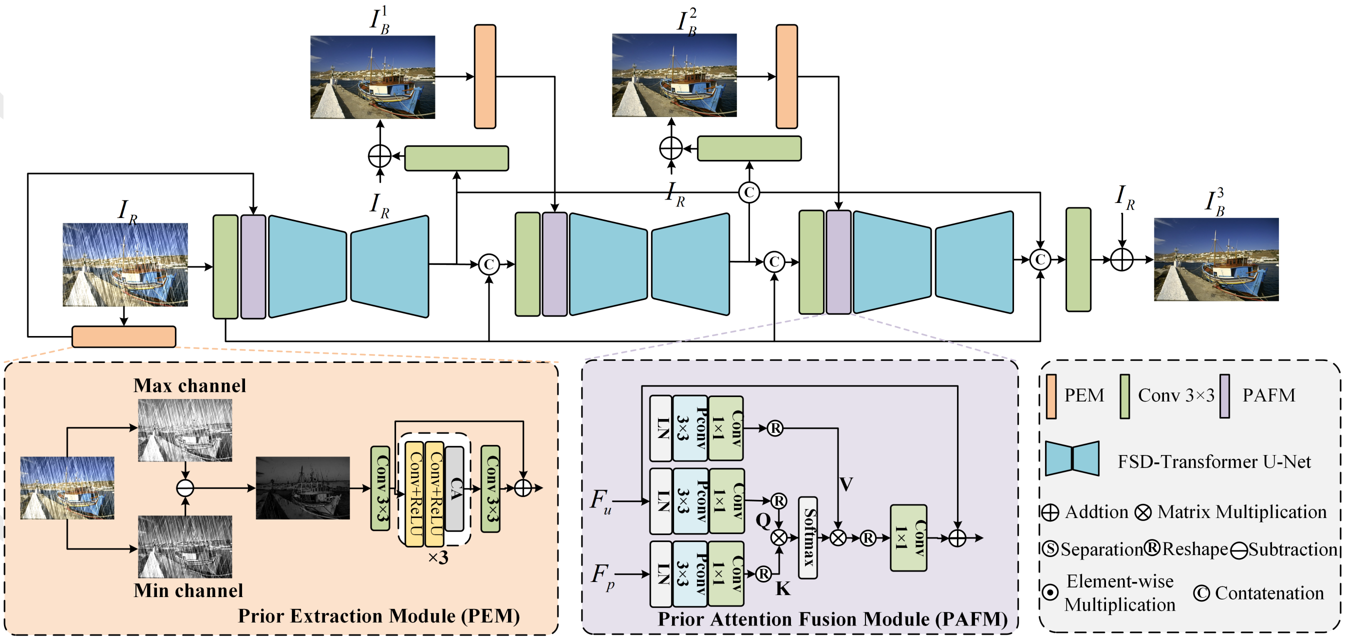


Figure 3: The overall architecture of FSDformer.

feature mapping layers containing one layer normalization operation, Partial Convolution (Pconv) [Chen *et al.*, 2023a], and one 1×1 convolutional layer to obtain the vectors Q, and V, and the background priori features are sent to one feature mapping layer to obtain the vector K. Then, Q, and K are used to generate the background prior attention matrix, which weights V to enhance background information. Finally, the weighted features are mapped to the original feature space and processed through a 1×1 convolution operation and a residual operation to obtain the enhanced shallow features F_{PA}. The operations of PAFM can be represented as follows.

$$\begin{aligned}\{Q,V\} = Reshape(Conv_{1\times1}(PConv(LN(F_u^i)))),\\ F_u^0 = F_0,\end{aligned} \tag{7}$$

$$\{K\} = Reshape(Conv_{1\times1}(PConv(LN(F_i^P)))), \tag{8}$$

$$F_{PA} = Conv_{1\times1}(softmax(\frac{QK^T}{\lambda}) \otimes v) + F_u^i, \tag{9}$$

where $LN(\cdot)$, $PConv(\cdot)$ and $Conv_{1\times1}(\cdot)$ denote the operations of layer normalization, 3×3 PConv, and 1×1 convolution, respectively. $Reshape(\cdot)$ denotes the operation of reshaping the input sequence, and $softmax(\cdot)$ denotes a Softmax activation layer. T denotes the matrix transpose operation, and $\otimes$ denotes matrix multiplication operation.

2.3 Fourier-spatial Dual Transformer (FSD-Transformer)

At present, Transformer has shown superior performance in rain tasks, but due to its focus on extracting global features and neglecting the learning of local features, there is still a problem of insufficient learning of rain streak structures. Therefore, we design an FSD-Transformer to achieve the extraction and construction of global and local features. FSD-Transformer mainly consists of local feature enhancement block (LFEB), Fourier-spatial dual attention (FDAB), and a Fourier enhancement gated block (FEGB). The structure of each block is as follows.

Local Feature Enhancement Block (LFEB)

To enhance the local feature learning ability of the transformer structure, LFEB is constructed as a multi-scale feature learning residual block, which extracts local features by utilizing convolution kernels with different receptive fields. The specific operations are as follows.

$$F_1 = GeLU(Conv_{1\times1(F_{in})}), \tag{10}$$

$$F_2 = GeLU(Conv_{1\times1}(PConv_{3\times3}(F_{in}))), \tag{11}$$

$$F_3 = GeLU(Conv_{1\times1}(PConv_{5\times5}(F_{in}))), \tag{12}$$

$$F_{Lf} = Conv_{1\times1}(Concat(F_1, F_2, F_3)) + F_{in}, \tag{13}$$

where $GeLU(\cdot)$ denotes the GeLU activation function, and $PConv_{3\times3}(\cdot)$ and $PConv_{5\times5}(\cdot)$ denotes Pconvs with convolution kernel sizes of 3×3 and 5×5 . F_1, F_2, F_3 represent the local features extracted by convolution kernels of sizes 1×1, 3×3, and 5×5, respectively. F_{in} and F_{Lf} denote the input and output of LFEB.

Fourier-spatial Dual Attention Block (FDAB)

To enhance the global feature learning ability of the transformer structure, FADB is designed to learn global features in both spatial and frequency domains by constructing a masked spatial self-attention block (MSSB) and a masked frequency self-attention block (MFSB), respectively. Firstly, 3×3 Pconv and 1×1 convolution operations are employed to achieve feature mapping and obtain feature vectors Q, K and V. In addition, to establish the global relationship of features in the

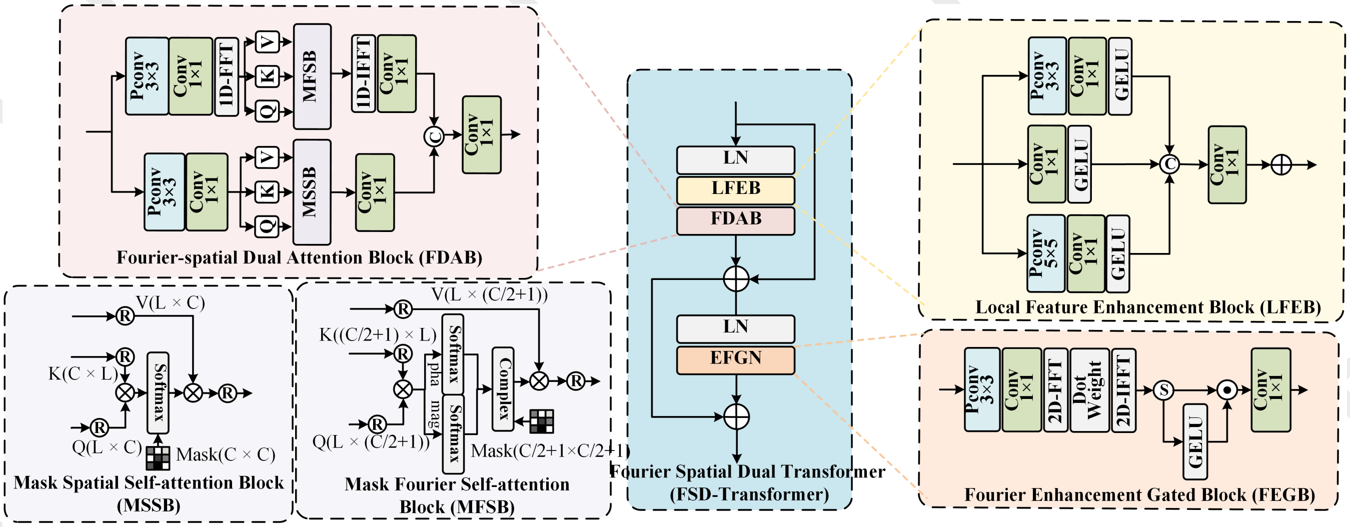


Figure 4: The structure of FSD-Transformer

frequency domain, a one-dimensional Fourier transform (1D-FFT) is employed.

$$\{Q_{sd}, K_{sd}, V_{sd}\} = Reshape(Conv_{1\times1}(PConv(F_{in}^{s}))), \tag{14}$$

$$\{Q_{fd}, k_{fd}, v_{fd}\} = 1D\_FFT(Reshape(Conv_{1\times1}(PConv(F_{Lf})))), \tag{15}$$

where Q_m, K_m and V_m denote feature vectors, $m = sd, fd$ denotes the spatial or frequency domain, and $1D\_FFT$ denotes the 1D-FFT operation.

Then, these feature vectors are fed into MSSB and MFSB to calculate the attention matrices, as shown in Figure 4. These two attention matrices are used to enhance spatial and frequency domain features, respectively. To improve the accuracy of feature extraction, two adaptive learning mask matrices $Mask_m$ are introduced to adjust the constructed self-attention matrices. The above operations are as follows.

$$EF_m = \big(Mask_m \odot \text{softmax}(Q_m K_m^T/\lambda)\big) \otimes V_m, \quad m=\{sd, fd\}, \tag{16}$$

where EF_m denotes the enhanced spatial and frequency features. $\odot$denotes the element-wise multiplication operation, and $\otimes$ denotes the matrix multiplication operation.

Finally, the enhanced frequency features EF_{fd} are transformed by inverse Fourier transform and integrated with the enhanced spatial features through a 1×1 convolution to obtain the output F_{FA} of FADB. The operations are as follows.

$$F_{FA} = Conv_{1\times1}(Concat(Conv_{1\times1}(1D\_IFFT(EF_{fd})), Conv_{1\times1}(EF_{sd})))), \tag{17}$$

where $1D\_IFFT$ denotes the one-dimensional inverse Fourier transform (1D-IFFT).

Fourier Enhancement Gated Block (FEGB)

FADB mainly enhances features by establishing the correlation of channel dimension features. To increase the correlation of global features in the spatial dimension, we designed a FEGB as shown in Figure 4 , which utilizes a two-dimensional Fourier transform to enhance the frequency features in the spatial dimension. Firstly, 3×3 Pconv and 1×1 convolution operations are employed to expand the channel number of the feature maps by twice. Then, learnable weights W are used to perform dot product operations on frequency features to achieve global feature enhancement. Next, a GeLU activation function is adopted to perform a gating operation in the spatial domain to further enhance the features. Finally, a 1×1 convolution operation is performed to achieve feature integration and channel dimensionality reduction.The above operations are as follows.

$$F_{ef} = IFFT_{2D}(FFT_{2D}(Conv_{1\times1}(PConv(F'))) \odot W), \tag{18}$$

$$F_{FE} = Conv_{1\times1}(GeLU(F_{ef} \odot F_{ef})), \tag{19}$$

where $FFT_{2D}(\cdot)$ and $IFFT_{2D}(\cdot)$ denote two-dimensional Fourier transform and inverse Fourier transform. F_{ef} denotes the enhanced global features in the frequency domain, and F'_{in} and F_{FE} denotes the input and output of FEGB.

2.4 Loss Function

To better guide the training of the network, we define a joint loss function consisting of Charbonnier loss L_c [Charbonnier *et al.*, 1994], Edge loss L_{edge} [Zamir *et al.*, 2021]] and Frequency Reconstruction loss L_f [30], which is defined as follows:

$$L_c = \sqrt{\|I_B - I_{GT}\|^2 + \varepsilon^2}, \tag{20}$$

$$L_{edge} = \sqrt{\|\Delta(I_B) - \Delta(I_{GT})\|^2 + \varepsilon^2}, \tag{21}$$

$$L_f = \|FFT(I_B) - FFT(I_{GT})\|_1, \tag{22}$$

where I_{GT} is the ground truth image, and the confidence level ε is set to 10^{-3}. In addition, to reconstruct more accurate features, L1 loss is used to constrain the output of each stage, which can be expressed as follows:

$$L_{st} = \sum_{k=1}^{2} \|I_k - I_{GT}\|_1, \tag{23}$$

where I_k is the output of the $k-th$ stage. Therefore, the joint loss function L_{total} is defined as follows:

$$L_{total} = L_c + \lambda_1 L_{edge} + \lambda_2 L_f + \lambda_3 L_1, \tag{24}$$

where the weighting factors λ_1, λ_2 and λ_3 are set to 0.05, 0.01 and 0.1, respectively.

3 Experiments

3.1 Experimental Settings

Datasets. All comparison experiments are conducted on a variety of public benchmark datasets, including Rain200H[Yang *et al.*, 2017], Rain200L[Yang *et al.*, 2017], DDN-Data[Luo *et al.*, 2015a], and DID-Data[Zhu *et al.*, 2017]. To further validate the generalization ability of the model, a large-scale real-world dataset (SPA-Data) [Wang *et al.*, 2019] including 638492 training image pairs and 1000 test image pairs [34], and the real dataset Internet-Data [Wang *et al.*, 2019] lacking reference images, are also employed to assess the efficacy of the various comparison methods.

Comparison Methods. To evaluate the performance of our method, we compared it with some SOTA methods, including two a priori-based traditional methods (DSC [Luo *et al.*, 2015b] and GMM [Li *et al.*, 2016]), eight CNN-based methods (DDN [Fu *et al.*, 2017], RESCAN [Li *et al.*, 2018], PReNet [Ren *et al.*, 2019], MSPFN [Jiang *et al.*, 2020], RCDNet [Wang *et al.*, 2020], MPRNet [Zamir *et al.*, 2021], DualGCN [Fu *et al.*, 2021] and SPDNet [Yi *et al.*, 2021]), as well as six Transformer-based methods (Uformer [Wang *et al.*, 2022], Restormer [Xiao *et al.*, 2022a], IDT [Xiao *et al.*, 2022b], DRSformer [Chen *et al.*, 2023b], MSTD [Chen *et al.*, 2024a], and NeRD [Chen *et al.*, 2024b]). To ensure fairness in the comparison, we adopted the same evaluation method used in [Chen *et al.*, 2024b].

Evaluation Metrics. To facilitate quantitative comparisons, two common evaluation metrics, PSNR [Huynh-Thu and Ghanbari, 2008] and SSIM [Wang *et al.*, 2004], are employed for the synthetic dataset and the real dataset SPA-Data. However, for the real dataset Internet-Data lacking reference images, two reference-free image quality estimation metrics, NIQE [Mittal *et al.*, 2012] and PIQE [Venkatanath *et al.*, 2015], are utilized to assess the performance of comparison methods.

Training Details. The proposed network is implemented using the PyTorch framework on a NVIDIA GeForce A6000 (48G). During the training process, the initial learning rate is 3×10^{-4}, and the Adam optimizer is used. The cosine annealing strategy is adopted to gradually reduce the learning rate, and the final learning rate is reduced to 1×10^{-6}. To increase the diversity of training samples, we performed random horizontal and vertical flipping operations on the training datasets for data augmentation. The patch size and batch size are set to 256× 256 and 6.

3.2 Experimental Results

Results on Synthetic Datasets. As shown in Table 1, our method achieves state-of-the-art PSNR/SSIM across all datasets. Figure 1 presents a qualitative comparison on the Rain200H dataset. It is obvious that the results obtained by other methods have edge blurring and significant artifacts, while our results have clearer edges and are closer to the GT image. This observation aligns with the quantitative values. This also demonstrates the effectiveness of our method.

Results on Real Datasets. The last column of Table 1 presents the objective results on the real dataset SPA. Similarly, our method achieved the best performance. In addition, to further validate the generalization of the comparison methods, we also conduct experiments on the Internet-Data dataset without reference images. Figs.2 and 7 show the subjective comparison results on the real datasets SPA and Internet Data, respectively. From the figures, it can be seen that our results have the least amount of residual rain streaks and retain more background information.

Table 2 presents the results of reference-free metrics on the dataset Internet-Data. The results indicate that our method achieves the optimal PIQE value, with NIQE value only 0.01 lower than MSDT. This also indicates that our method is also effective for real rainy images.

Model Efficiency. Table 3 compares computational complexity across Transformer-based methods. Our approach achieves the fewest parameters, second-lowest FLOPs, and fastest inference time on 256×256 images. Figure 8 further visualizes performance via a radar chart (parameters, FLOPs, runtime, PSNR, SSIM), demonstrating balanced superiority across all metrics.

3.3 Ablation Studies

To validate the effectiveness of various components in the model, we conducted extensive ablation experiments on the Rain200H dataset.

Effectiveness of Multi-stage Structure. To demonstrate the effectiveness of multi-stage architecture. The first stage in our network is referred to as the structure of Figure 1 (a), denoted as M1, and the first stage in which a rainy image is input in each layer is referred to as the structure of Figure 1 (b), denoted as M2. Our network is the multi-stage network in Figure 1 (C). To ensure fair comparison, the modules in all three architectures are the same. Table 4 presents the PSNR and SSIM values obtained by three architectures. The experimental results demonstrate that the progressive multi-stage architecture has achieved significant performance improvements.

Effectiveness of Each Module. To verify the effectiveness of each module in our model, ablation experiments were carried out and the experimental results are shown in the Table 5. In experiments, In the experiment, LFEB was directly removed, and PAFM was replaced by a Concatenation operation and a 1×1 convolution. For MSSB and MFSB, we only use spatial self-attention or Fourier self-attention to verify the effectiveness of both types of self-attention. Specifically, MSSB is replaced by MFSB, which means there are two

Datasets		Rain200H		Rain200L		DDN-Data		DID-Data		SPA-Data	
Metrics		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Prior-based methods	DSC	14.73	0.3815	27.16	0.8663	27.31	0.8373	24.24	0.8279	34.95	0.9416
	GMM	14.50	0.4164	28.66	0.8652	27.55	0.8479	25.81	0.8344	34.30	0.9428
CNN-based methods	DDN	26.05	0.8056	34.68	0.9671	30.00	0.9041	30.97	0.9116	36.16	0.9457
	RESCAN	26.75	0.8353	36.09	0.9697	31.94	0.9345	33.38	0.9417	38.11	0.9707
	PReNet	29.04	0.8991	37.80	0.9814	32.60	0.9459	33.17	0.9481	40.16	0.9816
	MSPFN	29.36	0.9034	38.58	0.9827	32.99	0.9333	33.72	0.955	43.43	0.9843
	RCDNet	30.24	0.9048	39.17	0.9885	33.04	0.9472	34.08	0.9532	43.36	0.9831
	MPRNet	30.67	0.9110	39.47	0.9825	33.10	0.9347	33.99	0.9590	43.64	0.9844
	DualGCN	31.15	0.9125	40.73	0.9886	33.01	0.9489	34.37	0.9620	44.18	0.9902
	SPDNet	31.28	0.9207	40.50	0.9875	33.15	0.9457	34.57	0.9560	43.20	0.9871
Transformer-based methods	Uformer	30.80	0.9105	40.20	0.9860	33.95	0.9545	35.02	0.9621	46.13	0.9913
	Restormer	32.00	0.9329	40.99	0.9890	34.20	0.9571	35.29	0.9641	47.98	0.9921
	IDT	32.10	0.9344	40.74	0.9884	33.84	0.9549	34.89	0.9623	47.35	0.9930
	DRSformer	32.17	0.9326	41.23	0.9894	34.35	0.9588	35.35	0.9646	48.54	0.9924
	MSDT	32.45	0.9379	41.75	0.9904	34.36	0.9593	35.37	0.9652	49.07	0.9926
	NeRD	32.40	0.9373	41.71	0.9903	34.45	0.9596	35.53	0.9659	49.58	**0.9940**
	Ours	**32.87**	**0.9421**	**42.05**	**0.9910**	**34.48**	**0.9604**	**35.65**	**0.9674**	**49.67**	**0.9940**

Table 1: Comparison of quantitative results on synthetic and real datasets. **bold** and underline indicate the best and second-best results.

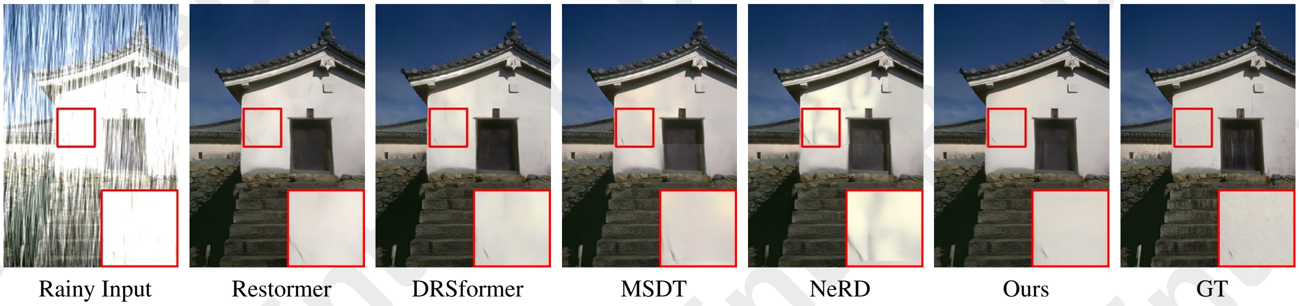


Figure 5: Visual quality comparison on the Rain200H dataset.

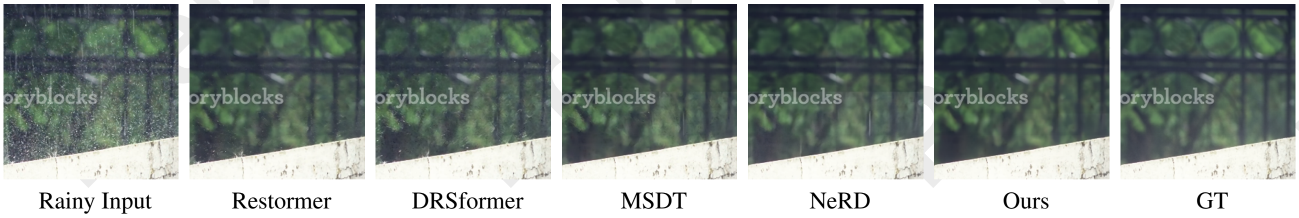


Figure 6: Visual quality comparison on the SPA-Data dataset.

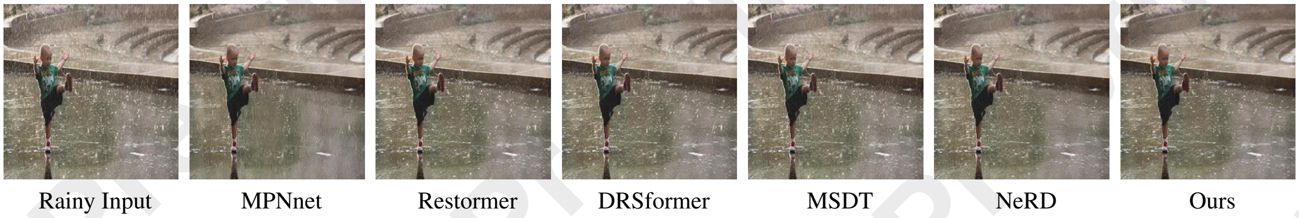


Figure 7: Visual quality comparison on the Internet-Data dataset.

Methods	SPDNet	Restormer	DRSformer	MSDT	NeRD	Ours
NIQE	3.69	3.70	3.68	**3.61**	3.64	3.62
PIQE	27.63	27.05	26.81	27.74	26.88	**26.08**

Table 2: Comparison of quantitative results on the internet-data dataset.

Methods	Restormer	IDT	DRSformer	MSDT	NeRD	Ours
Params(M)	26.12	16.41	33.70	16.60	22.86	**12.38**
FLOPs(G)	140.9	61.9	242.9	129.9	148.0	69.9
Runtimes(ms)	118.8	116.3	248.2	116.5	154.5	**78.4**

Table 3: Comparison of quantitative results on the internet-data dataset.

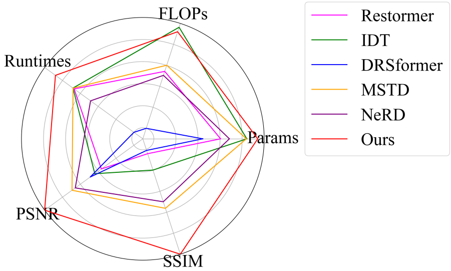
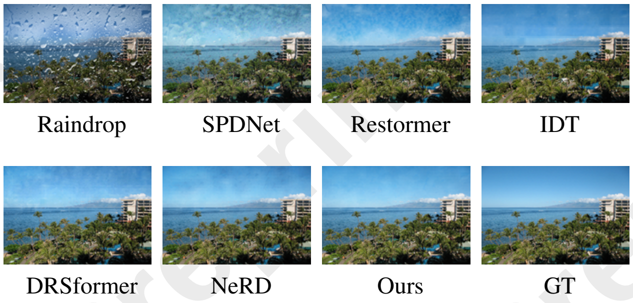


Figure 8: Performance comparison of three different architectures with increasing number of parameters.

MFSBs in FDAB. Similarly, MFSB is replaced by MSSB, which means there are two MSSBs in FDAB. The results show that removing any module affects the performance of the model, which also demonstrates the effectiveness of each module.

Effectiveness of Stage Number. To illustrate the impact of stage number on experimental results and computational costs, we conducted experiments using different stages such as 1, 2, 3, and 4, and the results are shown in Table 6. The results indicate that as the number of parameters increases, the performance of the model improves. When N is 4, the performance of this model only slightly improves compared to that of the three-stage model.


Raindrop SPDNet Restormer IDT

DRSformer NeRD Ours GT

Figure 9: Visual quality comparison of raindrop removal on the UAV-Rain1k dataset.

Methods	M_1	M_2	Ours
PSNR	32.08	32.38	**32.87**
SSIM	0.9305	0.9394	**0.9421**

Table 4: Effectiveness of Multi-stage.

Methods	LFEB	MSSB	MFSB	PAFM	PSNR
(a)		✓	✓	✓	32.61
(b)	✓		✓	✓	32.66
(c)	✓	✓		✓	32.56
(d)	✓	✓	✓		32.58
(e)	✓	✓	✓	✓	**32.87**

Table 5: Effectiveness of each module.

Methods	N=1	N=2	N=3	N=4
PSNR	31.27	32.49	32.87	**32.99**
Params(M)	4.13	8.25	12.38	16.59

Table 6: Effectiveness of stage number.

Methods	SPDNet	Restormer	IDT	DRSformer	NeRD	Ours
PSNR	22.54	24.78	22.47	24.93	25.57	**25.98**
SSIM	0.8594	0.9054	0.8957	0.9155	0.9219	**0.9277**

Table 7: Comparison of quantitative results on UAV-1K dataset.

3.4 Experiment on Removing Raindrops

To further validate the effectiveness of our model in removing raindrops, we conducted experiments on the UAV-1K [Chang *et al.*, 2024] dataset and compared the performance of various methods in removing raindrops. As shown in Table 7 and Fig. 9, our method achieves the highest PSNR/SSIM with minimal artifacts, outperforming existing approaches in visual quality and metrics.

4 Conclusion

This paper proposes a multi-stage image deraining network to gradually remove rain streaks. Each stage leverages PEM and PAFM to guide background reconstruction. A U-Net is constructed in each stage to achieve fine-grained extraction and reconstruction of background features at different scales by designing an FSD-Transformer, in order to obtain the rain removal result of each stage. FSD-Transformer is constructed to extract the global and local features in the spatial and frequency domains. Experiments demonstrate superior performance on both rain streaks and raindrops.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62072218 and No. 62261025), by the Natural Science Foundation of Tianjin (No. 24JCZDJC00130), and by the research funding provided by Cangzhou Institute of Tiangong University (No. TGCYY-Z-0303).

References

[Chang *et al.*, 2024] Wenhui Chang, Hongming Chen, Xin He, Xiang Chen, and Liangduo Shen. Uav-rain1k: A benchmark for raindrop removal from uav aerial imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15–22, 2024.

[Charbonnier *et al.*, 1994] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st international conference on image processing*, volume 2, pages 168–172. IEEE, 1994.

[Chen and Hsu, 2013] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *Proceedings of the IEEE international conference on computer vision*, pages 1968–1975, 2013.

[Chen *et al.*, 2023a] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: chasing higher flops for faster neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12021–12031, 2023.

[Chen *et al.*, 2023b] 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.

[Chen *et al.*, 2024a] Hongming Chen, Xiang Chen, Jiyang Lu, and Yufeng Li. Rethinking multi-scale representations in deep deraining transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1046–1053, 2024.

[Chen *et al.*, 2024b] Xiang Chen, Jinshan Pan, and Jiangxin Dong. Bidirectional multi-scale implicit neural representations for image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

[Fu *et al.*, 2017] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3855–3863, 2017.

[Fu *et al.*, 2021] Xueyang Fu, Qi Qi, Zheng-Jun Zha, Yurui Zhu, and Xinghao Ding. Rain streak removal via dual graph convolutional network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1352–1360, 2021.

[Huynh-Thu and Ghanbari, 2008] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

[Jiang *et al.*, 2020] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8346–8355, 2020.

[Kang *et al.*, 2011] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE transactions on image processing*, 21(4):1742–1755, 2011.

[Li *et al.*, 2016] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2736–2744, 2016.

[Li *et al.*, 2018] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 254–269, 2018.

[Luo *et al.*, 2015a] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE international conference on computer vision*, pages 3397–3405, 2015.

[Luo *et al.*, 2015b] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE international conference on computer vision*, pages 3397–3405, 2015.

[Mittal *et al.*, 2012] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

[Redmon, 2016] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[Ren *et al.*, 2019] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3937–3946, 2019.

[Venkatanath *et al.*, 2015] Narasimhan Venkatanath, D Praneeth, Maruthi Chandrasekhar Bh, Sumohana S Channappayya, and Swarup S Medasani. Blind image quality evaluation using perception based features. In *2015 twenty first national conference on communications (NCC)*, pages 1–6. IEEE, 2015.

[Viola and Jones, 2004] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57:137–154, 2004.

[Wang *et al.*, 2004] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[Wang *et al.*, 2019] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12270–12279, 2019.

[Wang *et al.*, 2020] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3103–3112, 2020.

[Wang *et al.*, 2022] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17683–17693, 2022.

[Xiao *et al.*, 2022a] Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image de-raining transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12978–12995, 2022.

[Xiao *et al.*, 2022b] Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image de-raining transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12978–12995, 2022.

[Yang *et al.*, 2017] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1357–1366, 2017.

[Yang *et al.*, 2019] Wenhan Yang, Robby T Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1377–1393, 2019.

[Yi *et al.*, 2021] Qiaosi Yi, Juncheng Li, Qinyan Dai, Faming Fang, Guixu Zhang, and Tieyong Zeng. Structure-preserving deraining with residue channel prior guidance. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4238–4247, 2021.

[Zamir *et al.*, 2021] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021.

[Zhang and Patel, 2018] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 695–704, 2018.

[Zhao *et al.*, 2017] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[Zhu *et al.*, 2017] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *Proceedings of the IEEE international conference on computer vision*, pages 2526–2534, 2017.