

# Bidirectional Multi-Scale Implicit Neural Representations for Image Deraining

Xiang Chen Jinshan Pan\* Jiangxin Dong  
Nanjing University of Science and Technology

## Abstract

*How to effectively explore multi-scale representations of rain streaks is important for image deraining. In contrast to existing Transformer-based methods that depend mostly on single-scale rain appearance, we develop an end-to-end multi-scale Transformer that leverages the potentially useful features in various scales to facilitate high-quality image reconstruction. To better explore the common degradation representations from spatially-varying rain streaks, we incorporate intra-scale implicit neural representations based on pixel coordinates with the degraded inputs in a closed-loop design, enabling the learned features to facilitate rain removal and improve the robustness of the model in complex scenarios. To ensure richer collaborative representation from different scales, we embed a simple yet effective inter-scale bidirectional feedback operation into our multi-scale Transformer by performing coarse-to-fine and fine-to-coarse information communication. Extensive experiments demonstrate that our approach, named as NeRD-Rain, performs favorably against the state-of-the-art ones on both synthetic and real-world benchmark datasets. The source code and trained models are available at <https://github.com/cschenxiang/NeRD-Rain>.*

## 1. Introduction

Recent years have witnessed significant progress in image deraining due to the development of numerous deep convolutional neural networks (CNNs) [15, 22, 51, 60, 63]. However, as the basic operation in CNNs, the convolution is spatially invariant and has limited receptive fields, which cannot effectively model the spatially-variant property and non-local structures of clear images [49, 62]. Moreover, simply increasing the network depth to obtain larger receptive fields does not always lead to better performance.

To alleviate this problem, several recent approaches utilize Transformers to solve single image deraining [5, 8, 20, 55, 56, 62], since Transformers can model the non-local information for better image restoration. Although these approaches achieve better performance than most of the CNN-

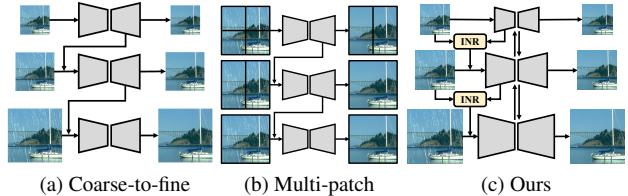


Figure 1. Illustration of the proposed approach and the currently existing multi-scale solutions. (a) coarse-to-fine scheme [22, 63]; (b) multi-patch scheme [61]; (c) our method. Compared to previous approaches, the method one integrates implicit neural representations (INR) into our bidirectional multi-scale model to form a closed-loop framework, which allows for better exploration of multi-scale information and modeling of complex rain streaks.

based ones, they mostly explore feature representations at a fixed image scale (*i.e.*, a single-input single-output architecture), while ignoring potentially useful information from other scales. As the rain effect decreases significantly at coarser image scales, exploring the multi-scale representation would facilitate the rain removal.

To this end, several approaches introduce the coarse-to-fine mechanism [12, 47] or multi-patch strategy [61] into deep neural networks to exploit multi-scale rain features. As shown in Figure 1, the decoder’s feature or derained image is initially estimated at a coarse scale and then used as the input at a finer scale for guidance. Although impressive performance has been achieved, these methods are less effective when handling complex and random rain streaks because these rain streaks cannot be removed by down-sampling operations and inaccurate estimation of a coarser scale will result in suboptimal restoration performance at finer scales. Despite the spatially-varying rain streaks exhibit a variety of scale properties (*e.g.*, size, shape, length, and density), the degraded rainy images tend to share some similar visual degradation characteristics (*i.e.*, common rain degradation representation) [52, 53, 56]. However, existing methods do not effectively model the common degradation as they usually rely on traditional representation forms that are sensitive to the input variation rather than capturing underlying implicit functions, which limits their performance on complex scenarios. Thus, it is of great interest to learn the underlying correlations among features to encode rain

\*Corresponding author.

appearance information from spatially-varying rain streaks.

Furthermore, we note that most existing multi-scale architectures [22, 26, 30, 47] utilize the features from coarser scales to facilitate the feature estimation at finer scales. However, if the features are not estimated correctly at coarser scales, the errors would affect the feature estimation at subsequent scales. Therefore, it is necessary to introduce a feedback mechanism to solve this problem.

In this paper, we develop an effective bidirectional multi-scale Transformer with implicit neural representations to better explore multi-scale information and model complex rain streaks. Considering that the rain effect varies at different image scales, we construct multiple unequal Transformer branches each for learning the scale-specific features for image deraining. Motivated by the recent success of implicit neural representations (INRs) that are able to encode an image as a continuous function, we further incorporate the INR between adjacent branches to learn common rain degradation representations from diverse degraded inputs so that the learned features are robust to complex and random rain streaks. To facilitate representing rain appearance at various scales, we employ two distinct coordinate-based multi-layer perceptrons (MLPs) (*i.e.*, one coarse and one fine feature grid) in INR for adaptively fitting complex rain characteristics. Furthermore, to improve the modeling capacity of the INR, we propose an intra-scale shared encoder to form a closed-loop framework. Note that the above mentioned two types of representation (*i.e.*, scale-specific and common rain ones) are able to complement each other.

To better build the interactions among features of different scales in a collaborative manner, we introduce a simple yet effective inter-scale bidirectional feedback mechanism into our proposed multi-scale Transformer. This enables the network to flexibly exchange information in both coarse-to-fine and fine-to-coarse flows, resulting in better robustness to variations in image content, such as changes in scale. Finally, we formulate the intra-scale INR branch and the inter-scale bidirectional branch into an end-to-end trainable image deraining network, named as NeRD-Rain. Experimental results demonstrate that our approach achieves favorable performance against state-of-the-art ones on the benchmark datasets, especially on real-world benchmarks.

The main contributions are summarized as follows:

- We design an effective multi-scale Transformer to generate high-quality deraining results with exploitation and utilization of multi-scale rain information.
- We introduce implicit neural representations to better learn common rain degradation features and show that it can help facilitate rain removal and enhance the robustness of the deraining model in complex scenes.
- We integrate a simple yet effective bidirectional feedback propagation operation into our multi-scale Transformer for better feature interaction across scales.

- Experimental results on both synthetic and real-world benchmarks demonstrate that our approach performs favorable performance against state-of-the-art ones.

## 2. Related Work

**Single image deraining.** Since the image deraining problem is ill-posed, conventional algorithms typically seek to impose some handcrafted priors [19, 24, 32, 35, 52, 64] on the clear images and rain components to make this problem well-posed. However, these methods fail to remove rain on complex real scenes when the assumptions do not hold. Afterwards, deep learning-based approaches [1, 7, 9] have outperformed early traditional algorithms and demonstrated decent restoration performance. As the development of this field, a wide array of network architectures and designs have been effectively explored to significantly boost the capacity of end-to-end model learning, *e.g.*, multi-scale [22, 30], multi-stage [31, 61], or multi-branch [53, 63] strategy.

Recently, researchers have made efforts to replace CNNs with Transformers as the fundamental structure for vision tasks [9, 13]. Driven by the great success of vision Transformer in modeling the non-local information, Transformer-based frameworks [5, 62] have emerged for better rain removal. For example, Xiao *et al.* [56] proposed an image deraining Transformer using spatial-based and window-based self-attention modules. Chen *et al.* [8] put forward a sparse Transformer to retain the most useful self-attention values for image reconstruction. Unfortunately, these approaches depend mostly on single-scale rain appearance, which limits their ability to fully explore multi-scale rain information. In this work, we investigate multi-scale representations in Transformer backbone for better boosting image deraining.

**Multi-scale representations.** The presence of rain streaks exhibits a noticeable degree of self-similarity, whether it is within the same scale or across different scales. This inherent property allows for the utilization of correlated features across scales in order to better represent rain information [9, 22, 30, 63]. In the context of CNNs, multi-scale representations have been explored for improving the image restoration performance, such as image pyramid [16], feature pyramid [37], coarse-to-fine mechanism [12, 47], and multi-patch scheme [61]. Recently, some studies [14] have investigated enforcing multi-scale design strategies to vision Transformers (ViT). For example, Chen *et al.* [3] formulated a cross-attention multi-scale Transformer for image classification. Lin *et al.* [33] developed a scale-aware modulation Transformer. In this work, we formulate a bidirectional multi-scale Transformer to solve image deraining.

**Implicit neural representation.** Implicit neural representation (INR) has emerged as a new compelling technique to represent continuous domain signals via coordinate-based multi-layer perceptrons (MLPs). Unlike explicit representations, which define the signal values at each point explicitly,

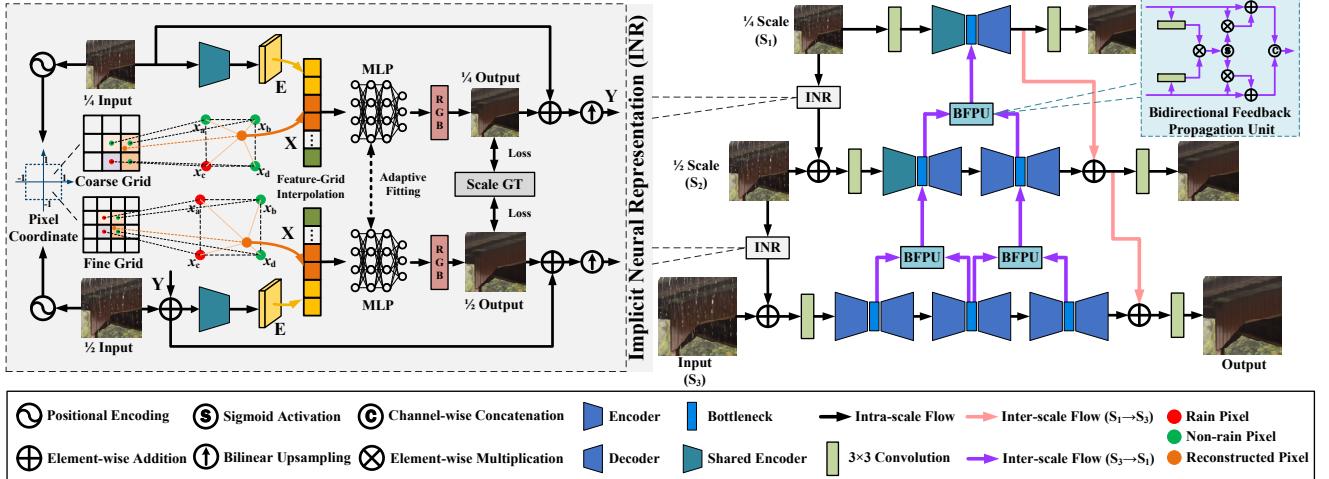


Figure 2. Overall architecture of the proposed bidirectional multi-scale Transformer with implicit neural representations (NeRD-Rain), which consists of intra-scale flows (*i.e.*, INR branch and unequal Transformer branch) and inter-scale flows (*i.e.*, coarse-to-fine and fine-to-coarse bidirectional branches). The proposed INR branch consists of two coordinated-based MLPs with coarse and fine feature grids. We construct an intra-scale shared encoder in the Transformer branch and INR branch, where two types of representation (*i.e.*, scale-specific and common rain ones) are able to complement each other. We formulate all the branches to form a closed-loop network architecture.

INR encodes the signal by learning a mapping from coordinates to signal values [44]. During the early period, it has been widely applied in various 3D vision tasks, *e.g.*, shape modeling [11], structure rendering [2], and scene reconstruction [21]. As a well-known approach, neural radiance fields (NeRF) [38] employs neural networks to represent complex 3D scenes by modeling the volumetric density and color at each point in space. Recent studies attempt to explore the potential of INRs for 2D images, *e.g.*, image compression [45], image reconstruction [39, 57], and arbitrary-scale image super-resolution [6, 10]. More recently, Yang *et al.* [58] utilized the controllable fitting capability of INR for low-light image enhancement problem. Quan *et al.* [41] proposed an INR-based inverse kernel prediction network to solve image defocus deblurring. Our work is inspired by this rapidly growing field and further demonstrates how to use implicit representation to better facilitate rain removal.

### 3. Proposed Method

To better explore multi-scale information and model complex rain streaks, we elaborately develop an effective bidirectional multi-scale Transformer with implicit neural representations (called NeRD-Rain), comprising an intra-scale INR branch and an inter-scale bidirectional branch. The former learns the underlying degradation representations from diverse rainy images, while the latter enables richer collaborative representations across different scales. Figure 2 summarizes the architecture of NeRD-Rain.

#### 3.1. Intra-scale INR branch

Given a rainy image  $I_{rain} \in \mathbb{R}^{H \times W \times 3}$ , where  $H \times W$  represents the spatial resolution of the input image, our method first uses bilinear interpolation to downsample the input im-

age into multi-scale versions (*i.e.*, 1/2 and 1/4). From the coarsest to the finest image scales, we designate the rescaled image at each scale as  $S_1$ ,  $S_2$ , and  $S_3$ , respectively. Different from previous multi-scale methods [22, 30, 47] that assign equal importance to the subnetworks of various scales, our approach incorporates the networks at finer scales with deeper architectures to handle spatially-varying rain streaks. At each scale, we propose unequal Transformer branches to perform deep feature extraction and generate a set of scale-specific outputs. Specifically, the NeRD-Rain at each scale (from  $S_1$  to  $S_3$ ) is equipped with one, two, and three UNets, respectively. Each UNet consists of a sequence of Transformer blocks [62]. Here, these UNets share the same network architecture but have independent weights [26].

In order to capture common rain degradation features, we further integrate an intra-scale INR branch into our multi-scale Transformer, which trains a multi-layer perceptron (MLP) by learning the following mapping function:

$$f_\theta : \mathbb{R}^2 \rightarrow \mathbb{R}^3, \quad (1)$$

where the input dimensions correspond to the  $(x, y)$  spatial coordinates of each pixel, while the output dimensions correspond to the  $(R, G, B)$  color channels of the pixels.

Specifically, we insert INRs between the inputs of adjacent Transformer branches to synchronously achieve rain reconstruction. Firstly, the input image  $I_{rain}$  is transformed to a feature map  $\mathbf{E} \in \mathbb{R}^{H \times W \times C}$  with a spatial resolution of  $H \times W$  pixels and  $C$  channels. Different from the INR in [58] which utilizes a separate encoder, we construct a shared encoder that interacts with the Transformer branch to form a compact closed-loop architecture, where these two types of representation (*i.e.*, scale-specific and common rain representation) can be utilized complementary to each

other. In addition, the position of each pixel from spatially-varying rain streaks is recorded in a relative coordinate set  $\mathbf{X} \in \mathbb{R}^{H \times W \times 2}$ , where the value ‘2’ represents horizontal and vertical coordinates. As suggested in [29, 38], we also adopt periodic spatial encoding to project the pixel coordinates  $\mathbf{X}$  into a higher dimensional space  $\mathbb{R}^{2L}$  for better recovering high-frequency details. The encoding procedure is formulated as:

$$\begin{aligned}\mathbf{X}' &= \gamma(\mathbf{X}), \\ \gamma(\mathbf{x}) &= [\sin(\mathbf{x}), \cos(\mathbf{x}), \dots, \sin(2^{L-1}\mathbf{x}), \cos(2^{L-1}\mathbf{x})],\end{aligned}\quad (2)$$

where  $\gamma(\cdot)$  represents a spatial encoding function.  $\mathbf{x} \in \mathbb{R}^{2L}$  is the coordinate value of  $\mathbf{X}$ , and it is normalized to lie within the range of  $[-1, 1]$ .  $L$  is a hyperparameter for determining dimension values. We set  $L = 4$  in our experiments.

Afterwards, a decoder is employed to predict RGB values of the output image by combining both  $\mathbf{E}$  and  $\mathbf{X}'$ . Here, our decoder consists of three-layer MLPs, with each layer having 256 hidden dimensions. Note that fitting an INR to reconstruct an image requires finding a set of parameters for the MLP  $f_\theta$  of a small size [43]. As a result, diverse types of rain streaks yield different sets of parameters, and it in turn means the MLP is adaptive to the common characteristics of all the degraded images. Similar to [10, 58], we calculate a weighted average of the predictions from the surrounding grids to obtain the RGB value ( $\mathbf{s} \in \mathbb{R}^3$ ) of the final reconstructed image, which can be viewed as an implicit neural interpolation process [6, 46]. This process is expressed as:

$$\begin{aligned}\mathbf{z} &= \mathbb{E}(I_{rain}), \\ \mathbf{s} &= \sum_{j \in \mathcal{J}} w_j f_\theta(\mathbf{z}_j, \mathbf{x}),\end{aligned}\quad (3)$$

where  $\mathbf{z}$  is a feature vector;  $\mathbb{E}$  represents a shared encoder;  $\mathcal{J} \in \mathbb{Z}^4$  is a set indices for four nearest (Euclidean distance) latent codes  $j$  around  $\mathbf{x}$ ;  $w_j$  denotes the local ensemble weight [29], satisfying  $\sum_j w_j = 1$ .

In experiments, we further find that this process can naturally facilitate rain removal without requiring any additional operations. Likewise, some studies [4, 25] also point out the low-pass filtering characteristics in INR. Due to the strong reflections caused by rain effect, pixels affected by rain tend to exhibit high intensity values, *i.e.*, white rain streaks [52]. Therefore, we attribute the deraining ability of INR to a basic fact that the intensity values of rain-affected pixels tend to surpass those of their neighboring non-rain pixels [52].

Instead of representing the image using INR at a fixed scale [6], we present a cascaded scale image representation for INR. Inspired by [18], our network trains two distinct MLPs, *i.e.*, one coarse and one fine feature grid. Through this sequential coarse-to-fine training, INR achieves more effective information transmission, naturally sharing information across scales. With all the above-mentioned designs, our INR branch can better learn common rain degradation

features so that the learned features are robust to complex and random rain streaks. These designs we consider yield performance improvements as we shall see in Section 5.

### 3.2. Inter-scale bidirectional branch

Although the intra-scale INR branch performs feature estimation from coarse to fine scales, it would affect the feature estimation for the subsequent scale when the features from the coarser scales are not estimated correctly. To overcome this problem, we introduce an inter-scale bidirectional branch into multi-scale Transformer, enabling both coarse-to-fine and fine-to-coarse feature propagation. Specifically, unlike using the complex and time-consuming LSTM [41], we formulate a simple yet effective bidirectional feedback propagation unit (BFPU) without adding much cost. Each BFPU takes the bottleneck layer features ( $F_a$  and  $F_b$ ) of two UNets at the current fine scales as inputs. The output of BFPU  $F_{out}$  is delivered to the bottleneck layer of the UNet at the previous coarse scale. In this way, the proposed BFPU can be formulated by:

$$\begin{aligned}F_{mid} &= \mathcal{S}(\text{Conv}_{3 \times 3}(F_a) \otimes \text{Conv}_{3 \times 3}(F_b)), \\ F_{out} &= [F_a + F_{mid} \otimes F_a, F_b + F_{mid} \otimes F_b],\end{aligned}\quad (4)$$

where  $\text{Conv}_{3 \times 3}$  is a  $3 \times 3$  convolution layer,  $\mathcal{S}$  denotes the Sigmoid function,  $\otimes$  is the element-wise multiplication, and  $[\cdot]$  represents the channel-wise concatenation.

With this design, the inter-scale bidirectional branch offers three-fold advantages: (1) it can make use of the complementary information from the subsequent (finer) scales to help image restoration at the current (coarser) scale, (2) it can perform feature propagation flow earlier without waiting for the derained results from previous scales, (3) it can be robust to variations in image content, such as changes in scale. We will show its effectiveness in Section 5.

### 3.3. Loss function

In order to jointly learn UNet-based traditional representations and INR-based continuous representations in a multi-scale manner, our network is trained end-to-end with a hybrid loss function. Following [12, 48], we employ the Charbonnier loss  $\mathcal{L}_{char}$  [61], the frequency loss  $\mathcal{L}_{freq}$  [23] and the edge loss  $\mathcal{L}_{edge}$  [22] to constrain scale-specific learning. Furthermore, we also employ a  $L_1$ -norm to avoid color shift during the prediction of RGB by INR. Based on one coarse and one fine feature grid, the total INR-related losses are calculated as follows:

$$\mathcal{L}_{inr} = \sum_{s=1}^2 \|\mathbf{I}_s - \mathbf{T}_s\|_1, \quad (5)$$

where  $\mathbf{I}_s$  and  $\mathbf{T}_s$  denote  $s$ -scale reconstructed image of INR and  $s$ -scale target ground-truth image. The proposed loss function  $\mathcal{L}_{total}$  for network training is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{char} + \alpha_1 \mathcal{L}_{freq} + \alpha_2 \mathcal{L}_{edge} + \alpha_3 \mathcal{L}_{inr}, \quad (6)$$

where the scalar weights  $\alpha_1, \alpha_2$  and  $\alpha_3$  are empirically set to 0.01, 0.05 and 0.1, respectively.

Table 1. Quantitative evaluations of the proposed approach against state-of-the-art methods on five commonly used benchmark datasets. Our NeRD-Rain achieves higher quantitative results, especially advances state-of-the-art by **1.04 dB** on the real benchmark, SPA-Data.

Datasets		Rain200L [59]		Rain200H [59]		DID-Data [63]		DDN-Data [15]		SPA-Data [51]	
Metrics		PSNR	SSIM								
Prior-based methods	DSC [35]	27.16	0.8663	14.73	0.3815	24.24	0.8279	27.31	0.8373	34.95	0.9416
	GMM [32]	28.66	0.8652	14.50	0.4164	25.81	0.8344	27.55	0.8479	34.30	0.9428
CNN-based methods	DDN [15]	34.68	0.9671	26.05	0.8056	30.97	0.9116	30.00	0.9041	36.16	0.9457
	RESCAN [31]	36.09	0.9697	26.75	0.8353	33.38	0.9417	31.94	0.9345	38.11	0.9707
	PReNet [42]	37.80	0.9814	29.04	0.8991	33.17	0.9481	32.60	0.9459	40.16	0.9816
	MSPFN [22]	38.58	0.9827	29.36	0.9034	33.72	0.9550	32.99	0.9333	43.43	0.9843
	RCDNet [50]	39.17	0.9885	30.24	0.9048	34.08	0.9532	33.04	0.9472	43.36	0.9831
	MPRNet [61]	39.47	0.9825	30.67	0.9110	33.99	0.9590	33.10	0.9347	43.64	0.9844
	DualGCN [17]	40.73	0.9886	31.15	0.9125	34.37	0.9620	33.01	0.9489	44.18	0.9902
	SPDNet [60]	40.50	0.9875	31.28	0.9207	34.57	0.9560	33.15	0.9457	43.20	0.9871
Transformer-based methods	Uformer [55]	40.20	0.9860	30.80	0.9105	35.02	0.9621	33.95	0.9545	46.13	0.9913
	Restormer [62]	40.99	0.9890	32.00	0.9329	35.29	0.9641	34.20	0.9571	47.98	0.9921
	IDT [56]	40.74	0.9884	32.10	0.9344	34.89	0.9623	33.84	0.9549	47.35	0.9930
	DRSformer [8]	41.23	0.9894	32.17	0.9326	35.35	0.9646	34.35	0.9588	48.54	0.9924
	NeRD-Rain-S	41.30	0.9895	32.06	0.9315	35.36	0.9647	34.25	0.9578	48.90	0.9936
NeRD-Rain		<b>41.71</b>	<b>0.9903</b>	<b>32.40</b>	<b>0.9373</b>	<b>35.53</b>	<b>0.9659</b>	<b>34.45</b>	<b>0.9596</b>	<b>49.58</b>	<b>0.9940</b>

## 4. Experimental Results

We first discuss the experimental settings of our proposed NeRD-Rain. Then we evaluate the effectiveness of our approach on both synthetic and real-world datasets. More results are included in the supplemental material.

### 4.1. Experimental settings

**Datasets and metrics.** We evaluate our approach on four commonly used synthetic benchmarks (*i.e.*, Rain200L [59], Rain200H [59], DID-Data [63], and DDN-Data [15]), and two recent real-world datasets (*i.e.*, SPA-Data [51] and RE-RAIN [9]). We follow the protocols of these benchmarks for training and testing. To evaluate the quality of each derained image, we use PSNR [40] and SSIM [54] as the evaluation metrics when the ground truth images are available, and calculate them based on the Y channel of the YCbCr color space, following the previous works [8, 9].

**Implementation details.** In our proposed NeRD-Rain, each Transformer-based UNet adopts a 3-level encoder-decoder architecture. From level-1 to level-3, the number of Transformer blocks is set to [2, 3, 3], the number of self-attention heads in [62] is set to [1, 2, 4], and the number of channels is set to [48, 96, 192]. We introduce another variant, NeRD-Rain-S, by modifying the number of feature channels to [32, 64, 128]. We implement our method based on the PyTorch framework and train it from scratch using a machine with one NVIDIA GeForce RTX 3090 GPU. During the training, we use the Adam optimizer [27]. The patch size is set to be 256 × 256 pixels and the batch size is set to be 1. We train the SPA-Data dataset for 10 epochs, the DID-Data and DDN-Data datasets for 200 epochs, and the Rain200L and Rain200H datasets for 600 epochs. The same data augmentation method [36] is adopted. For training on the Rain200H dataset, the learning rate is initialized

as  $2 \times 10^{-4}$ , while for other benchmarks, it is initialized as  $1 \times 10^{-4}$ . The final learning rate is gradually decreased to  $1 \times 10^{-6}$  using a cosine annealing scheme [34].

### 4.2. Comparisons with the state of the art

We compare our method with prior-based methods (DSC [35] and GMM [32]), CNN-based approaches (DDN [15], RESCAN [31], PReNet [42], MSPFN [22], RCDNet [50], MPRNet [61], DualGCN [17] and SPDNet [60]), and recent Transformer-based methods (Uformer [55], Restormer [62], IDT [56], and DRSformer [8]).

**Evaluations on synthetic datasets.** Table 1 summarizes the quantitative evaluation results on the above-mentioned synthetic datasets [15, 59, 63], where the derained images by our method have higher PSNR and SSIM values. For example, the PSNR values of our approach is at least 0.48 dB higher than DRSformer [8] on the Rain200L benchmark.

We further show some visual results on the Rain200H dataset in Figure 3. The CNN-based methods [17, 22, 50, 60, 61] do not recover clear images in heavy rainy scenarios. The Transformer-based methods [8, 55, 56, 62] are able to model the global contexts for image deraining. However, some main structures, *e.g.*, slender cable, are not recovered well. Compared to existing Transformer-based methods that depend on single-scale rain appearance, our developed multi-scale Transformer is able to explore multi-scale representations of rain streaks, and generates clearer images with fine details and structures.

**Evaluations on real-world datasets.** We further evaluate our method on the challenging SPA-Data dataset [51]. As provided in the last column of Table 1, our method outperforms the DRSformer method [8] by 1.04 dB on the SPA-Data dataset. This indicates that our method can effectively deal with diverse types of spatially-varying real rain streaks.

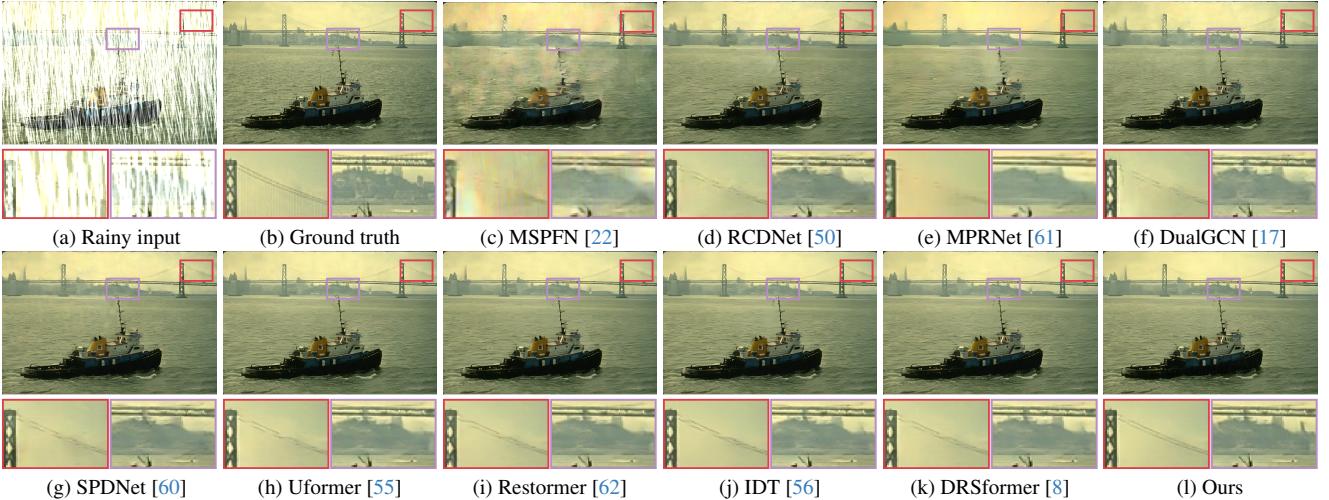


Figure 3. Derained results on the Rain200H dataset [59]. Compared with the derained results in (c)-(k), our method recovers a high-quality image with clearer details. Zooming in the figures offers a better view at the deraining capability.

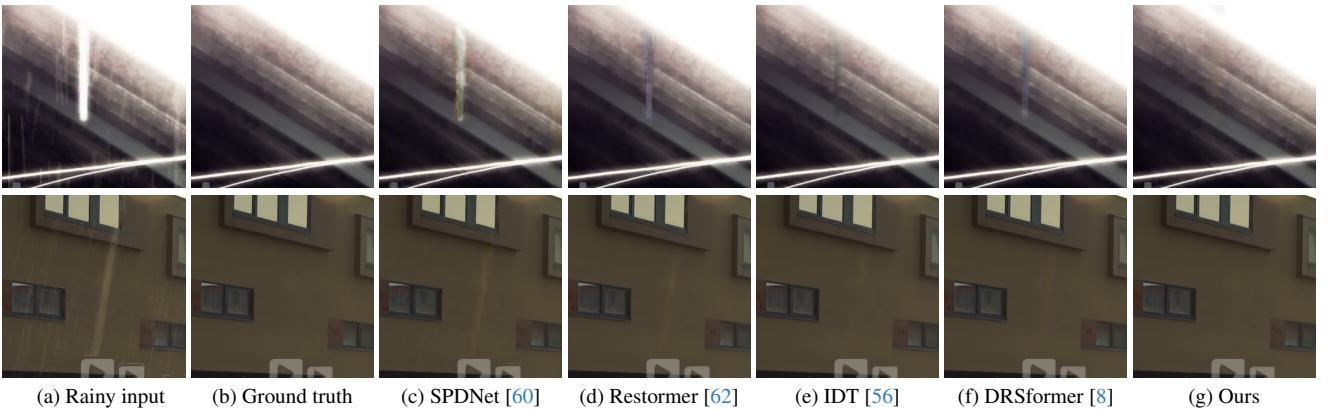


Figure 4. Derained results on the SPA-Data [51] dataset. Compared with the derained results in (c)-(f), our method recovers clearer images.

Table 2. Comparisons of model complexity against state-of-the-art methods. The size of the test image is  $256 \times 256$  pixels. “#FLOPs” and “#Params” represent FLOPs (in G) and the number of trainable parameters (in M), respectively.

Methods	MSPFN [22]	IPT [5]	Uformer [55]	Restormer [62]
#FLOPs (G)	595.5	1188	45.9	174.7
#Params (M)	13.35	115.5	50.88	26.12
Methods	IDT [56]	DRSformer [8]	Ours-S	Ours
#FLOPs (G)	61.9	242.9	79.2	156.3
#Params (M)	16.41	33.65	10.53	22.89

Figure 4 shows some visual comparisons of the evaluated methods, where our method generates better derained images. In contrast, the recovery results of other methods still contain some undesired rain streaks residual.

We also evaluate our method using real captured rainy images from the RE-RAIN dataset [9], where ground truths are not available. Figure 5 shows that most deep models are sensitive to spatially-long rain streaks and leave behind residual rain effect. On the contrary, our method effectively removes random rain streaks and achieves better recovery results, indicating its ability to generalize well on real data.

**Model complexity.** We evaluate the model complexity of

our method and state-of-the-art ones in terms of FLOPs and model parameters. Table 2 shows that our model, NeRD-Rain-S, has lower FLOPs value and fewer parameters while achieving competitive performance as shown in Table 1.

## 5. Analysis and Discussion

To better understand how the proposed approach solves image deraining, we examine the effect of the main component by conducting ablation studies. We train our method and all the alternative baselines using the same settings for fairness.

**Effect of multi-scale configuration.** Our developed multi-scale Transformer focuses on incorporating finer scales with deeper architectures to better remove spatially-varying rain streaks. To demonstrate the effectiveness of this formulation, we analyze the effect of different multi-scale configurations on the Rain200L dataset. As shown in Table 4, these variants differ in the number of UNets at different scales while sharing the same network architecture for the UNets. Compared to treating each scale equally (*i.e.*, M222), our method (*i.e.*, M123) can better reconstruct scale-specific feature and improve potential image restoration quality.



Figure 5. Derained results on a real-world rainy image from [9]. Compared with the derained results in (b)-(f), our method removes most rain streaks and recovers a clearer image. Zooming in the figures offers a better view at the deraining capability.

Table 3. Ablation analysis on different variants of INR in our method, including four aspects: position, feature-grid, operation, and feature encoder. Here, all ablation models adopt the backbone of bidirectional multi-scale Transformer (BMT).

Methods	Implicit Neural Representation (INR)							Backbone	Metrics			
	Position		Feature grid		Operation							
	Within branch	Adjacent branch	Fixed-scale	Multi-scale	Position encoding	Interpolation	Shared encoder	Separate encoder				
(a)	✗	✗	✗	✗	✗	✗	✗	✗	✓	41.40 0.9896		
(b)	✓	✗	✗	✓	✓	✓	✗	✓	✓	41.51 0.9897		
(c)	✗	✓	✗	✓	✓	✓	✓	✗	✓	41.71 0.9903		
(d)	✗	✓	✓	✗	✓	✓	✓	✗	✓	41.47 0.9897		
(e)	✗	✓	✗	✓	✗	✓	✓	✗	✓	41.50 0.9899		
(f)	✗	✓	✗	✓	✓	✓	✓	✗	✓	41.58 0.9900		
(g)	✗	✓	✗	✓	✓	✓	✗	✓	✓	41.62 0.9901		

Table 4. Ablation analysis on the multi-scale configuration using different numbers of UNets. Here,  $S_1$ ,  $S_2$ , and  $S_3$  represent 1/4, 1/2 and full image scale, respectively.

Methods	$S_1$	$S_2$	$S_3$	PSNR	SSIM
M222	2	2	2	41.42	0.9896
M321	3	2	1	40.70	0.9882
M023	0	2	3	41.63	0.9901
M123 (Ours)	1	2	3	<b>41.71</b>	<b>0.9903</b>

**Effectiveness of INR branch.** To analyze the effectiveness of INR branch, we conduct ablation experiments based on different variants in Table 3. All variants are trained on the Rain200L dataset. We compare with the baseline without using INR (*i.e.*, model (a)). In contrast, our model (c) exhibits better performance, especially in real scenes where the capability for effective rain removal is more pronounced (Figure 6(b) and (d), see the supplemental material for more examples). The approach without INR is sensitive to rain streaks of different scales, while our method successfully removes diverse rain streaks. This confirms that the learned representations from our INR better facilitates rain removal.

To understand the effect of such INR, we first visualize the output of INR from our network, and pixel value distribution of images. In Figure 8, we find that INR reduces the high intensity pixel values of rainy images and generates rain-free images. As mentioned in Section 3.1, we attribute this natural ability to the fact that INR tends to bias towards learning low-frequency image contents [29], while rain occupies the high-frequency part of the image.

Then, we further analyze the effect of implicit interpolation and position encoding in INR. Figure 7(b) shows that

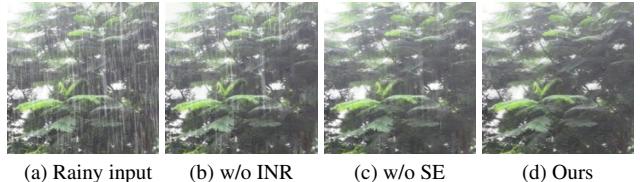


Figure 6. Ablation qualitative comparison on a real-world rainy image. ‘SE’ denotes a shared encoder in our method.

INR fails to remove undesired high-frequency rain streaks without using implicit interpolation operations. Meanwhile, the output of INR without using position encoding are prone to losing background details, as shown in Figure 7(c). All in all, the former aims to find the local latent embedding to reconstruct RGB values of the clear image, while the latter focuses on encoding coordinates with the frequency information to recover fine-grained background details.

Next, we also evaluate the effect of the position of INR in our framework. Compared to embedding the INR within the Transformer branch (*i.e.*, model (b) in Table 3), our method inserts INR between adjacent scales to naturally share information across scales, thereby improving quantitative and qualitative results (Figure 7(d)). Furthermore, we note that our method with multi-scale feature grid boosts the representation capacity of INR, which benefits from observing rain appearance at different resolutions (Figure 7(e) and (f)).

Finally, we demonstrate the effectiveness of shared encoder on image deraining. Here, we construct separate encoders for each INR as a comparison. For fair comparison, we set the same network architecture for these encoders as our method. By comparing model (c) and (g) in Table 3,

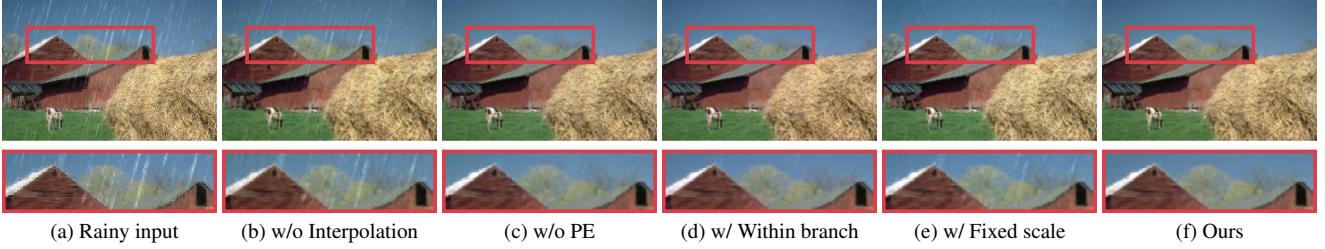


Figure 7. Visual quality comparison of output results of INR in different variants. Our INR branch generates a clearer image using all the designs we consider, which further provides better guidance for the input of the next scale. “PE”denotes the position encoding operation.

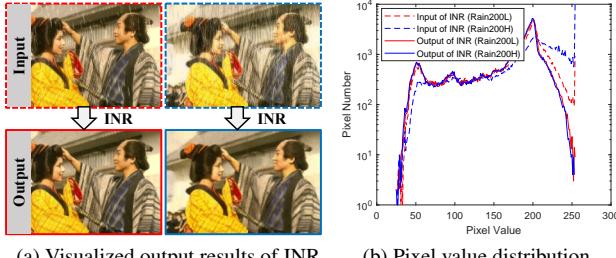


Figure 8. Comparisons between the input images and the results of INR. One can see that INR reduces the high intensity pixel values of white rain streaks and reconstructs the latent rain-free images.

our method achieves higher quantitative results with fewer network parameters. Figure 6(c) and (d) also demonstrate that using the shared encoder generates much clearer images. In contrast, the method with the separate encoder still has residual rain streaks in the recovery results. This further indicates that our formulation is robust to complex real-world scenarios. The utilization of shared encoders forms a more compact closed-loop framework, enabling the learned degradation representations to better facilitate rain removal.

**Effectiveness of bidirectional branch.** The BFPU in the bidirectional branch is used to better explore bidirectional information in our multi-scale Transformer for better image restoration. To demonstrate the effectiveness of this branch, we remove this component and investigate its influence in Table 5. We note that the BFPU achieves a PSNR gain of 0.10 dB over unidirectional propagation (*i.e.*, w/o BFPU) on the Rain200H dataset. Compared to direct feature concatenation, our bidirectional branch can dynamically aggregate richer features to facilitate image restoration. Figure 9 also shows that our method generates much clearer details.

**Extension to various architectures..** We extend our network architecture to CNN-based U-Net to demonstrate the scalability of the proposed framework. Here, we choose the U-Net module in MPRNet [61] as the baseline. Table 6 reports the results trained on the Rain200L benchmark. Our method still achieves competitive performance, indicating the effectiveness of our method in various architectures.

**Limitations and failure cases.** Although our NeRD-Rain achieves favorable performance on several image deraining

Table 5. Ablation quantitative comparison on the proposed BFPU.

Methods	w/o BFPU	w/ Concat	w/ BFPU (Ours)
PSNR / SSIM	32.30 / 0.9346	32.31 / 0.9360	<b>32.40 / 0.9373</b>

Table 6. Extension to CNN-based U-Net [61] in our NeRD-Rain.

Methods	MPRNet [61]	DRSformer [8]	Ours
PSNR / SSIM	39.47 / 0.9825	41.23 / <b>0.9894</b>	<b>41.34 / 0.9893</b>

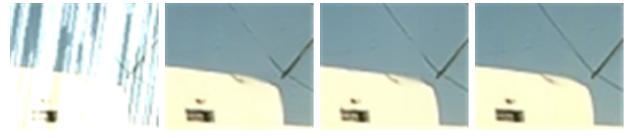


Figure 9. Ablation qualitative comparison on the proposed BFPU.

benchmarks, its training time is relatively longer compared to other multi-scale architectures. This is mainly due to the time-consuming optimization process involved in INR. Future work will apply the model pruning or early stopping scheme to improve training speed while maintaining performance [28]. Furthermore, our method fails to handle the veiling effect (*i.e.*, mist) in complex rainy environments.

## 6. Concluding Remarks

We have presented an effective multi-scale Transformer network for single image deraining. To better explore common rain degradation features, we incorporate coordinated-based implicit neural representation between adjacent scales, enabling the learned features can better facilitate rain removal and improve the robustness of the model in complex scenarios. To enhance the interactions among features of different scales in a collaborative manner, we also introduce a simple yet effective bidirectional feedback propagation operation into our multi-scale Transformer by performing coarse-to-fine and fine-to-coarse information communication. By formulating the proposed method into an end-to-end trainable model, we show that it performs favorably against the state-of-the-art methods on both synthetic and real benchmarks.

**Acknowledgements.** This work has been partly supported by the National Natural Science Foundation of China (Nos. U22B2049, 62272233, 62332010), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (No. KYCX23\_0486).

## References

- [1] Yunhao Ba, Howard Zhang, Ethan Yang, Akira Suzuki, Arnold Pfahl, Chethan Chinder Chandrappa, Celso M de Melo, Suya You, Stefano Soatto, Alex Wong, et al. Not just streaks: Towards ground truth for single image deraining. In *ECCV*, pages 723–740, 2022. [2](#)
- [2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, pages 5855–5864, 2021. [3](#)
- [3] Chun-Fu Richard Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *ICCV*, pages 357–366, 2021. [2](#)
- [4] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *NeurIPS*, 34:21557–21568, 2021. [4](#)
- [5] 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*, pages 12299–12310, 2021. [1, 2, 6](#)
- [6] Hao-Wei Chen, Yu-Syuan Xu, Min-Fong Hong, Yi-Min Tsai, Hsien-Kai Kuo, and Chun-Yi Lee. Cascaded local implicit transformer for arbitrary-scale super-resolution. In *CVPR*, pages 18257–18267, 2023. [3, 4](#)
- [7] Xiang Chen, Jinshan Pan, Kui Jiang, Yufeng Li, Yufeng Huang, Caihua Kong, Longgang Dai, and Zhentao Fan. Unpaired deep image deraining using dual contrastive learning. In *CVPR*, pages 2017–2026, 2022. [2](#)
- [8] Xiang Chen, Hao Li, Mingqiang Li, and Jinshan Pan. Learning a sparse transformer network for effective image deraining. In *CVPR*, pages 5896–5905, 2023. [1, 2, 5, 6, 7, 8](#)
- [9] Xiang Chen, Jinshan Pan, Jiangxin Dong, and Jinhui Tang. Towards unified deep image deraining: A survey and a new benchmark. *arXiv preprint arXiv:2310.03535*, 2023. [2, 5, 6, 7](#)
- [10] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *CVPR*, pages 8628–8638, 2021. [3, 4](#)
- [11] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, pages 5939–5948, 2019. [3](#)
- [12] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *ICCV*, pages 4641–4650, 2021. [1, 2, 4](#)
- [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. *ICLR*, 2020. [2](#)
- [14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6824–6835, 2021. [2](#)
- [15] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *CVPR*, pages 3855–3863, 2017. [1, 5](#)
- [16] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. Lightweight pyramid networks for image de-raining. *IEEE TNNLS*, 31(6):1794–1807, 2019. [2](#)
- [17] Xueyang Fu, Qi Qi, Zheng-Jun Zha, Yurui Zhu, and Xinghao Ding. Rain streak removal via dual graph convolutional network. In *AAAI*, pages 1352–1360, 2021. [5, 6](#)
- [18] Sharath Girish, Abhinav Shrivastava, and Kamal Gupta. Shacira: Scalable hash-grid compression for implicit neural representations. In *CVPR*, pages 17513–17524, 2023. [4](#)
- [19] Shuhang Gu, Deyu Meng, Wangmeng Zuo, and Lei Zhang. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In *ICCV*, pages 1708–1716, 2017. [2](#)
- [20] Yun Guo, Xueyao Xiao, Yi Chang, Shumin Deng, and Luxin Yan. From sky to the ground: A large-scale benchmark and simple baseline towards real rain removal. In *CVPR*, pages 12097–12107, 2023. [1](#)
- [21] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *CVPR*, pages 6001–6010, 2020. [3](#)
- [22] 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 *CVPR*, pages 8346–8355, 2020. [1, 2, 3, 4, 5, 6](#)
- [23] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *ICCV*, pages 13919–13929, 2021. [4](#)
- [24] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE TIP*, 21(4):1742–1755, 2011. [2](#)
- [25] Chaewon Kim, Jaeho Lee, and Jinwoo Shin. Zero-shot blind image denoising via implicit neural representations. *arXiv preprint arXiv:2204.02405*, 2022. [4](#)
- [26] Kiyeon Kim, Seungyong Lee, and Sunghyun Cho. Mssnet: Multi-scale-stage network for single image deblurring. In *ECCV*, pages 524–539, 2022. [2, 3](#)
- [27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. [5](#)
- [28] Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. Ffcv: Accelerating training by removing data bottlenecks. In *CVPR*, pages 12011–12020, 2023. [8](#)
- [29] Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *CVPR*, pages 1929–1938, 2022. [4, 7](#)
- [30] Pengpeng Li, Jiyu Jin, Guiyue Jin, Lei Fan, Xiao Gao, Tianyu Song, and Xiang Chen. Deep scale-space mining network for single image deraining. In *CVPRW*, pages 4276–4285, 2022. [2, 3](#)
- [31] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, pages 254–269, 2018. [2, 5](#)

- [32] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *CVPR*, pages 2736–2744, 2016. [2](#), [5](#)
- [33] Weifeng Lin, Ziheng Wu, Jiayu Chen, Jun Huang, and Lianwen Jin. Scale-aware modulation meet transformer. In *ICCV*, 2023. [2](#)
- [34] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. [5](#)
- [35] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015. [2](#), [5](#)
- [36] Xintian Mao, Yiming Liu, Fengze Liu, Qingli Li, Wei Shen, and Yan Wang. Intriguing findings of frequency selection for image deblurring. In *AAAI*, pages 1905–1913, 2023. [5](#)
- [37] Yiqun Mei, Yuchen Fan, Yulun Zhang, Jiahui Yu, Yuqian Zhou, Ding Liu, Yun Fu, Thomas S Huang, and Humphrey Shi. Pyramid attention network for image restoration. *IJCV*, pages 1–19, 2023. [2](#)
- [38] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. [3](#), [4](#)
- [39] Seonghyeon Nam, Marcus A Brubaker, and Michael S Brown. Neural image representations for multi-image fusion and layer separation. In *ECCV*, pages 216–232, 2022. [3](#)
- [40] Huynh-Thu Q. and Ghanbari M. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13): 800–801, 2008. [5](#)
- [41] Yuhui Quan, Xin Yao, and Hui Ji. Single image defocus deblurring via implicit neural inverse kernels. In *ICCV*, pages 12600–12610, 2023. [3](#), [4](#)
- [42] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *CVPR*, pages 3937–3946, 2019. [5](#)
- [43] Fernando Rivas-Manzaneque, Angela Ribeiro, and Orlando Avila-García. Ice: Implicit coordinate encoder for multiple image neural representation. *IEEE TIP*, 2023. [4](#)
- [44] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representation. In *ECCV*, pages 318–333, 2022. [3](#)
- [45] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *ECCV*, pages 74–91, 2022. [3](#)
- [46] Jiaxiang Tang, Xiaokang Chen, and Gang Zeng. Joint implicit image function for guided depth super-resolution. In *ACM MM*, pages 4390–4399, 2021. [4](#)
- [47] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jia-ya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, pages 8174–8182, 2018. [1](#), [2](#), [3](#)
- [48] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *CVPR*, pages 5769–5780, 2022. [4](#)
- [49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. [1](#)
- [50] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, pages 3103–3112, 2020. [5](#), [6](#)
- [51] 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 *CVPR*, pages 12270–12279, 2019. [1](#), [5](#), [6](#)
- [52] Yinglong Wang, Shuaicheng Liu, Chen Chen, and Bing Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE TIP*, 26(8):3936–3950, 2017. [1](#), [2](#), [4](#)
- [53] Yinglong Wang, Yibing Song, Chao Ma, and Bing Zeng. Rethinking image deraining via rain streaks and vapors. In *ECCV*, pages 367–382, 2020. [1](#), [2](#)
- [54] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. [5](#)
- [55] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, pages 17683–17693, 2022. [1](#), [5](#), [6](#), [7](#)
- [56] Jie Xiao, Xueyang Fu, Aiping Liu, Feng Wu, and Zheng-Jun Zha. Image de-raining transformer. *IEEE TPAMI*, 2022. [1](#), [2](#), [5](#), [6](#), [7](#)
- [57] Dejia Xu, Peihao Wang, Yifan Jiang, Zhiwen Fan, and Zhangyang Wang. Signal processing for implicit neural representations. *NeurIPS*, 35:13404–13418, 2022. [3](#)
- [58] Shuzhou Yang, Moxuan Ding, Yanmin Wu, Zihan Li, and Jian Zhang. Implicit neural representation for cooperative low-light image enhancement. In *ICCV*, 2023. [3](#), [4](#)
- [59] 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 *CVPR*, pages 1357–1366, 2017. [5](#), [6](#)
- [60] Qiaosi Yi, Juncheng Li, Qinyan Dai, Faming Fang, Guixu Zhang, and Tieyong Zeng. Structure-preserving deraining with residue channel prior guidance. In *ICCV*, pages 4238–4247, 2021. [1](#), [5](#), [6](#), [7](#)
- [61] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, pages 14821–14831, 2021. [1](#), [2](#), [4](#), [5](#), [6](#), [8](#)
- [62] 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*, pages 5728–5739, 2022. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#)
- [63] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, pages 695–704, 2018. [1](#), [2](#), [5](#)
- [64] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *ICCV*, pages 2526–2534, 2017. [2](#)