

# Vision Transformers for Single Image Dehazing

Yuda Song, Zhuqing He, Hui Qian, and Xin Du.

**Abstract**—Image dehazing is a representative low-level vision task that estimates latent haze-free images from hazy images. In recent years, convolutional neural network-based methods have dominated image dehazing. However, vision Transformers, which has recently made a breakthrough in high-level vision tasks, has not brought new dimensions to image dehazing. We start with the popular Swin Transformer and find that several of its key designs are unsuitable for image dehazing. To this end, we propose DehazeFormer, which consists of various improvements, such as the modified normalization layer, activation function, and spatial information aggregation scheme. We train multiple variants of DehazeFormer on various datasets to demonstrate its effectiveness. Specifically, on the most frequently used SOTS indoor set, our small model outperforms FFA-Net with only 25% #Param and 5% computational cost. To the best of our knowledge, our large model is the first method with the PSNR over 40 dB on the SOTS indoor set, dramatically outperforming the previous state-of-the-art methods. We also collect a large-scale realistic remote sensing dehazing dataset for evaluating the method’s capability to remove highly non-homogeneous haze. We share our code and dataset at <https://github.com/IDKiro/DehazeFormer>.

**Index Terms**—Image Processing, Image Dehazing, Deep Learning, Vision Transformer.

## I. INTRODUCTION

HAZE is a common atmospheric phenomenon that can impair daily life and machine vision systems. The presence of haze reduces the scene’s visibility and affects people’s judgment of the object, and thick haze can even affect traffic safety. For computer vision, haze degrades the quality of the captured image in most cases. It can impact the model’s reliability in high-level vision tasks, further mislead machine systems, such as autonomous driving. All these make image dehazing a meaningful low-level vision task.

Image dehazing aims to estimate the latent haze-free image from the observed hazy image. For the single image dehazing problem, there is a popular model [1–3] to characterize the degradation process for hazy images:

$$I = J(x)t(x) + A(1 - t(x)), \quad (1)$$

where  $I$  is the captured hazy image,  $J$  is the latent haze-free image,  $A$  is the global atmospheric light, and  $t$  is the medium transmission map. And the transmission can be expressed as

$$t(x) = e^{-\beta d(x)}, \quad (2)$$

where  $\beta$  is the scattering coefficient of the atmosphere, and  $d$  is the scene depth. As can be seen, image dehazing is a typically

Manuscript received XXXX 00, 0000; accepted XXXX 00, 0000. Date of publication XXXX 00, 0000; date of current version XXXX 00, 0000. The associate editor coordinating the review of this manuscript and approving it for publication was XXXX. (Yuda Song and Zhuqing He contributed equally to this work.) (Corresponding authors: Xin Du.)

Yuda Song, Zhuqing He, Hui Qian, and Xin Du are with Zhejiang University, Hangzhou 310027, China (e-mail: duxin@zju.edu.cn)

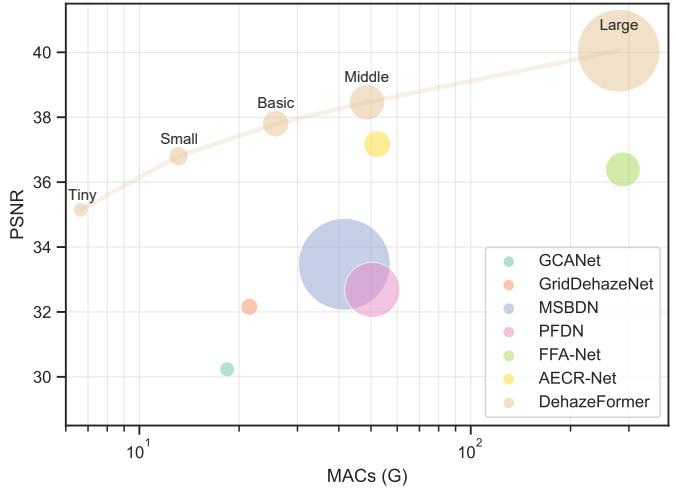


Fig. 1. Comparison of DehazeFormer with other image dehazing methods on the SOTS indoor set. The size of the dots indicates the #Param of the method, and MACs are shown with logarithmic axis.

ill-posed problem, and early image dehazing methods tend to constrain the solution space with priors [4–7]. They generally estimate  $A$  and  $t(x)$  separately to lower the complexity of the problem and then use Eq.(1) to derive the results. These prior-based methods can produce images with good visibility. However, these images are often visibly different from haze-free images, and artifacts may be introduced in regions that do not satisfy the priors.

In recent years, deep learning has made a big hit in computer vision, and researchers have proposed a large number of image dehazing methods based on deep convolutional neural networks (CNNs) [8–21]. With a sufficient number of synthetic image pairs, these methods can achieve superior performance over prior-based methods. Earlier CNN-based methods [8–10] also estimate  $A$  and  $t(x)$  separately, where  $t(x)$  is supervised using the transmission map used in synthesizing the dataset. And current methods [13–20] prefer to predict the latent haze-free image or the residuals of the haze-free image versus the hazy image since it tends to achieve better performance. Very recently, ViT [22] outperformed almost all CNN architectures in high-level vision tasks using plain Transformer architecture. Subsequently, many modified architectures [23–40] have been proposed, and vision Transformer is challenging the dominance of CNNs in high-level vision tasks. So many works have demonstrated the effectiveness of vision Transformers, but there is still no Transformer-based image dehazing method that defeats the state-of-the-art image dehazing networks. In this work, we propose an image dehazing Transformer dubbed DehazeFormer, which is inspired by Swin Transformer [30]. It dramatically surpasses these CNN-based methods.

We find that the LayerNorm [41] and GELU [42] commonly used in vision Transformers harm the image dehazing performance. Specifically, the LayerNorm used in vision Transformer normalizes the tokens corresponding to the image patches separately, resulting in the loss of the relativity between the patches. Hence, we remove the normalization layer preceded by the multi-layer perceptron (MLP) and propose RescaleNorm to replace LayerNorm. RescaleNorm performs normalization on the entire feature map and reintroduces the mean and variance of the feature map lost after normalization. Besides, SiLU / Swish [43] and GELU work well in high-level vision tasks, but ReLU [44] works better than them in image dehazing. We believe this is because the nonlinearities they introduce are not easily inverted when decoding. We argue that image dehazing requires not only that the network encodes highly expressive features but also that these features are easily recovered to image domain signals.

Swin Transformer uses window partitioning with cyclic shift to efficiently aggregate local features. But we find that the cyclic shift is suboptimal for image edge regions in image dehazing. Specifically, the cyclic shift should use masked multi-head self-attention (MHSA) to prevent unreasonable spatial aggregation, making the windows in the edge regions smaller. We consider that aggregating information within a small window brings instability, which can bias the network's training. Thus we propose a shifted window partitioning scheme based on reflection padding and cropping, which allows MHSA to discard the mask and achieve a constant window size. We also find that the aggregation weights of MHSA are always positive, which makes it behave like a low-pass filter [29]. Because the aggregation weights of MHSA are dynamic, all-positive, and normalized, we believe that static, learnable, and unconstrained aggregation weights are helpful to complement the MHSA, while the convolution meets this criterion.

Furthermore, we propose a prior-based soft reconstruction module that outperforms global residual learning and a multi-scale feature map fusion module based on SKNet [45] to replace concatenation fusion. Finally, we build multiple U-Net-like image dehazing Transformers using the proposed modules. Our experiments show that DehazeFormer can substantially outperform contemporaneous methods with lower overhead. Fig. 1 shows the comparison of DehazeFormer with other image dehazing methods on the SOTS indoor set. Our small model defeats the FFA-Net [18] with only 25% #Param and 5% computational cost. Our base model is lower in overhead but better in performance than the previous state-of-the-art method, AECR-Net [19]. To the best of our knowledge, our large model is the first method over 40 dB, substantially outperforming contemporaneous methods.

There are non-homogeneous image dehazing datasets collected using professional haze machines [46], but they are too small and far from the non-homogeneous haze that would be present in natural scenes. Instead, we tend to collect remote sensing image dehazing datasets since highly non-homogeneous haze is prevalent in remote sensing images. We take into account the wavelength, *etc.*, on the spatial distribution of the haze and then synthesize a large-scale realistic remote sensing image dehazing dataset.

## II. RELATED WORKS

### A. Image Dehazing

Early single-image dehazing methods were generally based on the handcraft priors, such as dark channel prior (DCP) [4], color attenuation prior (CAP) [6], color-lines [5], and hazelines [7]. These prior-based methods usually yield images with good visibility. However, because these priors are based on empirical statistics, these dehazing methods tend to output unrealistic results when the scenes do not satisfy these priors. With the rapid development of deep learning, learning-based dehazing methods have dominated in recent years. DehazeNet [8] and MSCNN [9] are the pioneers in applying CNNs for image dehazing. They learn to estimate  $t$  and obtain the result together with  $A$  estimated by the conventional method. After that, DCPDN [10] uses two sub-networks to estimate  $t$  and  $A$  respectively, while GFN [12] estimates the fusion coefficient maps for the three predefined image operations. AOD-Net [11], on the other hand, rewrites Eq.(1) so that the network needs to estimate only one component. GridDehazeNet [13] proposes that learning to restore the image is better than estimating  $t$ , because the latter will fall into suboptimal solutions. And most recent works [14–20] tend to estimate the haze-free image or the residual between the haze-free image and the hazy image.

Since the dehazing performance of the learning-based methods dramatically depends on the quality and size of the dataset, several datasets have been proposed. These dehazing datasets are divided into two main categories: real datasets [46–49] and synthetic datasets [50–52]. Real datasets use real haze produced by professional haze machines to generate real hazy images. Synthetic datasets generally use Eq.(1) to synthesize the corresponding hazy images with haze-free images and depth maps. Although the real datasets seem to be more attractive, it is difficult to obtain enough image pairs, and the distribution of the haze produced by the haze machine still differs significantly from the real haze. Hence, most methods tend to use synthetic datasets for training and testing. In contrast to these datasets, this paper presents a new synthetic remote sensing image dehazing dataset named RS-Haze for evaluating the method's capability to remove highly non-homogeneous haze. RS-Haze is larger and more realistic than previous datasets [53–56], taking into account sensor characteristics, haze distribution and particle size, wavelengths of light, and other factors that are overlooked.

### B. Vision Transformers

CNN has dominated most computer vision tasks for years, while recently, the Vision Transformer (ViT) [57] architectures show the capability of replacing CNNs. ViT pioneered the direct application of the Transformer architecture [22], which projects images into token sequences via patch-wise linear embedding. The shortcomings of the original ViT are its weak inductive bias and the quadratic computational cost. To this end, PVT [23] uses the pyramid architecture to introduce multi-scale inductive bias and downsamples the key and value to reduce the computational cost. T2T-ViT [24] uses the unfolding operation just like CNNs for tokenization, and it

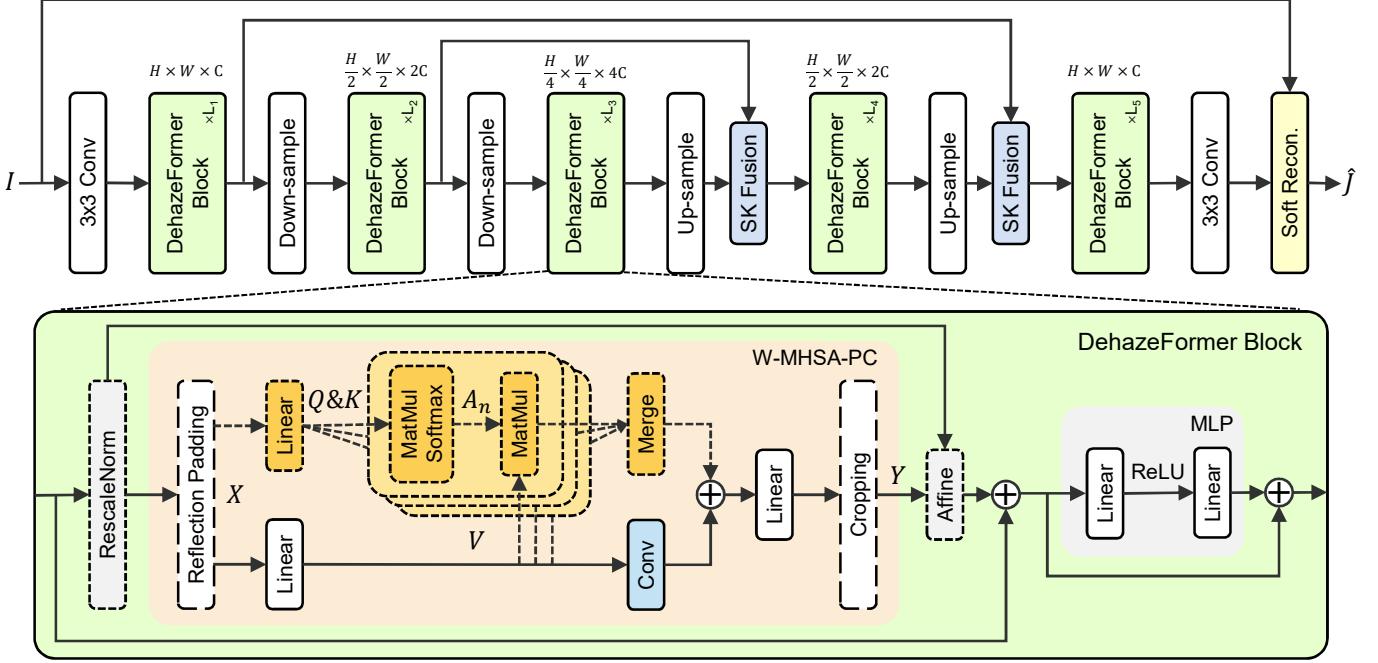


Fig. 2. DehazeFormer is a modified 5-stage U-Net, whose convolutional blocks are replaced by our DehazeFormer blocks. The components illustrated with dashed boxes in the DehazeFormer block indicate they are optional. The SK fusion and soft reconstruction layers are proposed to replace the original concatenation fusion and global residual. The input size is  $H \times W$ , and the size of feature maps in each stage is shown below the DehazeFormer block.

uses the Performer [25] to lower the computational cost. Besides, some works [26–29] employ convolution in the early stages to introduce the inductive bias. Swin Transformer [30] partitions tokens into windows and performs self-attention within a window to keep the linear computational cost. It employs the cyclic shift scheme to bridge windows so that adjacent blocks adopt different window partitions. Since then, many follow-ups to Swin Transformer have been proposed. For example, some methods bridge windows by reshaping the tensor [31–34]; while some methods bridge windows by using tokens with global receptive fields as proxies [35–37]; and others use modified window partitioning schemes [38–40]. Our DehazeFormer can be considered as a combination of Swin Transformer and U-Net [58], but with several critical modifications for image dehazing.

There are also some variants of Swin Transformer for low-level vision tasks. SwinIR [59] is one of the pioneers to employ Swin Transformer in low-level vision tasks, which builds a large residual block consisting of stacked Swin Transformer layers and a subsequent convolutional layer. Uformer [60] uses Swin Transformer blocks to build a U-Net-like network and inserted depth-wise convolution (DWConv) [61] in the feed-forward network (FFN) like LocalViT [62]. However, we found that they perform very poorly in image dehazing. We attribute this to the fact that they inherit the normalization layer, window partitioning scheme, and activation function from the original Swin Transformer. There are a few ViT-based dehazing networks proposed, such as HyLoG-ViT [63] and TransWeather [64]. However, HyLoG-ViT does not show convincing performance, while TransWeather aims to use a DETR-like framework [65] to handle multiple weather conditions simultaneously.

### III. DEHAZEFORMER

#### A. Overall

DehazeFormer’s network architecture is based on the popular Swin Transformer [30], but incorporates several improvements to compensate for the deficiencies of the original Swin Transformer when dealing with image dehazing. Fig. 2 shows the overall architecture of the DehazeFormer. Given the image pair  $\{I(x), J(x)\}$ , we only compute the  $L_1$  loss to train the DehazeFormers.

First, we briefly review the Swin Transformer. Given an input feature map  $X \in \mathbb{R}^{b \times h \times w \times c}$ , we project  $X$  to  $Q, K, V$  (query, key, value) using linear layers and group tokens using window partitioning. Swin Transformer applies MHSA within the window, and the window partitioning of adjacent blocks is different. For simplicity, the following  $Q, K, V \in \mathbb{R}^{b \times l \times d}$  correspond to a single window & header, where  $l$  is the tokens number in a window and  $d$  is the dimension. Thus the self attention is computed by

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(QK^T/\sqrt{d} + B\right)V, \quad (3)$$

where  $B$  is the relative position bias term. A linear layer follows it to project the output of the attention.

Our proposed DehazeFormer block differs from the original Swin Transformer block in the normalization layer, the nonlinear activation function, and the spatial information aggregation scheme, detailed in the subsequent subsections. Besides the DehazeFormer block, the SK fusion and soft reconstruction layers are proposed to replace the concatenation fusion layer and global residual learning.

The SK fusion layer is inspired by SKNet [45], it is designed to fuse multiple branches using channel attention. Let the two

feature maps be  $x_1$  and  $x_2$ , we first use a linear layer  $f(\cdot)$  to project  $x_1$  to  $\hat{x}_1$ . We use the global average pooling  $\text{GAP}(\cdot)$ , MLP (Linear-ReLU-Linear)  $\mathcal{F}_{MLP}(\cdot)$ , softmax function and split operation to obtain the fusion weights:

$$\{a_1, a_2\} = \text{Split}(\text{Softmax}(\mathcal{F}_{MLP}(\text{GAP}(\hat{x}_1 + x_2))). \quad (4)$$

We use the weights  $\{a_1, a_2\}$  to fuse  $\hat{x}_1, x_2$  with an additional short residual via  $y = a_1\hat{x}_1 + a_2x_2 + x_2$ .

Current image dehazing networks generally predict the reconstructed image  $\hat{J}(x)$  or global residual  $R(x) = \hat{J}(x) - I(x)$ . We consider it beneficial to introduce priors, provided that there are no strong constraints since the degradation model is an approximation. We rewrite Eq.(1) as

$$J(x) = K(x)I(x) + B(x) + I(x), \quad (5)$$

where  $K(x) = 1/t(x) - 1$  and  $B(x) = -(1/t(x) - 1)A$ . We drive the network to predict  $O \in \mathbb{R}^{h \times w \times 4}$ , and split  $O$  into  $K \in \mathbb{R}^{h \times w \times 1}$  and  $B \in \mathbb{R}^{h \times w \times 3}$ . As a result, the network architecture softly constrains the relationship between  $K(x)$  and  $B(x)$ . This weak prior allows the network to degenerate to predict global residuals (*i.e.*,  $K(x) = \mathbf{0}$ ,  $B(x) = R(x)$ ). For convenience, we refer to Eq.(5) as soft reconstruction.

### B. Rescale Layer Normalization

The normalization layer plays a vital role in neural network architecture since it stabilizes the network's training. However, we find that LayerNorm [41], which Transformers commonly use, may be unsuitable for image dehazing.

We first review the formulation of LayerNorm used by Transformers. Assume that the shape of the feature map  $x \in \mathbb{R}^{b \times n \times c}$ , where  $n = h \times w$  (*i.e.*, height and width), the normalization process can be expressed as:

$$\hat{x}_i = \frac{x_i - \mu_i}{\sigma_i} \cdot \gamma_i + \beta_i. \quad (6)$$

Here  $\mu$  and  $\sigma$  denote the mean and standard deviation,  $\gamma$  and  $\beta$  are learned scaling factor and bias, and  $i = (i_b, i_n, i_c)$  denotes the index. In LayerNorm,  $\mu$  and  $\sigma$  are computed along the  $c$ -axis, making  $\mu, \sigma \in \mathbb{R}^{b \times n}$ . We believe that the mean and standard deviation are correlated with brightness and contrast for images, so the relative brightness and contrast between image patches are somehow discarded after LayerNorm. To this end, we compute  $\mu$  and  $\sigma$  along the  $(n, c)$ -axes, leading to  $\mu, \sigma \in \mathbb{R}^b$ . We note this normalization method is the LayerNorm more commonly used in CNNs, referred to as LayerNorm $^\dagger$  in this paper.

We conduct a simple experiment to show the negative effects of LayerNorm as shown in Fig. 3. Specifically, we build autoencoders using only patch embedding, normalization, and patch reconstruction layers. We train these autoencoders to reconstruct a single input image. Without global residuals, learning identity mappings is not a trivial task [66]. When LayerNorm is inserted, we can clearly see the block artifacts appearing in the reconstructed image. Because this autoencoder does not involve interactions between patches, it can only memorize the statistics of the sky region at the expense of the rich-texture region. By changing LayerNorm to LayerNorm $^\dagger$ , we largely overcome its negative effects.

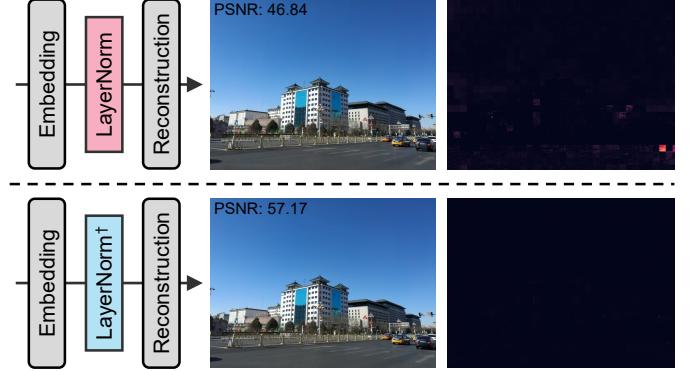


Fig. 3. Simple autoencoders for analyzing the normalization methods. From left to right, there are the autoencoders' architectures, output images, and error maps, where the error is scaled by 8x for better viewing. The embedding layer and reconstruction layer are linear layers with patch-wise tensor reshaping.

However, LayerNorm $^\dagger$  still discards the mean and standard deviation of the feature map. So we propose the Rescale Layer Normalization (RescaleNorm), which is built based on LayerNorm $^\dagger$ , but the mean and standard deviation computed are saved and introduced at the end of the residual block. Specifically, we first fetch the  $\mu, \sigma \in \mathbb{R}^{b \times 1 \times 1}$ , and normalize the input feature map  $x$  to  $\hat{x}$  via Eq.(6). We then use the main block  $\mathbf{F}(\cdot)$  to process  $\hat{x}$  to obtain the output  $\hat{y}$ . We use two linear layers with weights  $W_\gamma, W_\beta \in \mathbb{R}^{1 \times c}$  and biases  $B_\gamma, B_\beta \in \mathbb{R}^{1 \times 1 \times c}$  to transform  $\mu$  and  $\sigma$  via  $\{\gamma, \beta\} = \{\sigma W_\gamma + B_\gamma, \mu W_\beta + B_\beta\}$ , where  $\gamma, \beta \in \mathbb{R}^{b \times 1 \times c}$ . To accelerate convergence, we initialize  $B_\gamma$  and  $B_\beta$  to  $\mathbf{1}$  and  $\mathbf{0}$ . We inject  $\gamma$  and  $\beta$  into  $\hat{y}$  to reintroduce the mean and standard deviation. Therefore, RescaleNorm can be formulated as:

$$y = \mathbf{F}\left(\frac{x - \mu}{\sigma} \cdot \gamma + \beta\right) \cdot (\sigma W_\gamma + B_\gamma) + (\mu W_\beta + B_\beta). \quad (7)$$

Compared to BatchNorm [67], LayerNorm is not a cheap operation. It needs to compute the mean and standard deviation during inference instead of using the running estimates tracked on the training set. Therefore, we remove the normalization layer before the MLP, as we find that this hardly worsens the method's performance.

### C. Nonlinear Activation Function with Simple Inversal

GELU performs better than ReLU in high-level tasks [43, 68, 69]. However, GELU is much less used than ReLU [44] and LeakyReLU [70] in low-level vision tasks. Although some recent Transformer-based image processing networks inherit GELU [59, 60], in our experiments, ReLU and LeakyReLU still perform better than GELU in image dehazing. We believe that GELU does not work in the image dehazing task because it is not easily inverted. If we consider GELU as an image filter, it causes the gradient reversal problem because of its non-monotonicity. Unlike high-level vision tasks, the feature maps in image dehazing would be decoded into images, resulting in the reversal gradients introduced by GELU to react in the output image.

Comparing GELU and ReLU, another reason for GELU's inferior performance is its stronger nonlinearity since it is

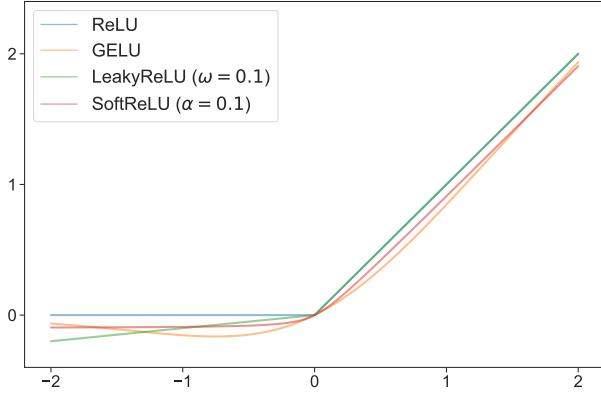


Fig. 4. The ReLU, GELU, LeakyReLU ( $\omega = 0.1$ ), and SoftReLU ( $\alpha = 0.1$ ).

more complicated than piece-wise linear functions. Hence we propose SoftReLU, which is a simple smooth approximation to the ReLU as an excess between GELU and ReLU:

$$\text{SoftReLU}(x) = \frac{x + \sqrt{x^2 + \alpha^2} - \alpha}{2}. \quad (8)$$

where  $\alpha$  is a shape parameter. In particular, SoftReLU is equivalent to ReLU when we set  $\alpha = 0$ . To mimic GELU, we set  $\alpha = 0.1$  in our experiments.

Fig. 4 illustrates a comparison of the SoftReLU with other nonlinear activation functions. We perform ablation studies on activation functions and find that LeakyReLU performs similarly to the ReLU, better than SoftReLU and GELU, while SoftReLU is better than GELU. Therefore, we believe that the nonlinear activation function's invertibility is essential for image dehazing networks.

#### D. Shifted Window Partitioning with Reflection Padding

Swin Transformer uses cyclic shift with masked MHSA to implement efficient batch computation for shifted window partitioning. Because of the mask, the window size at the edge of the image is smaller than the set window size. For high-level vision tasks, the object of the image is often in the center of the image, making the edge pixels of the image contribute little to the result. For image dehazing, image edges are as important as image centers. A small window size leads to a smaller number of tokens in the window, which biases the network's training. We consider that the network's performance can be improved by keeping the window size of the image edges the same as the set window size.

To avoid introducing unreasonable inter-patch interactions, we propose to use reflection padding to achieve efficient batch computation for shifted window partitioning, as Fig. 4 illustrated. Swin Transformer's original paper mentions how to use padding to implement batch computation. However, its proposed padding-based scheme is equivalent to the cyclic shift since the masked MHSA will still be employed. Unlike Swin Transformer, we use reflection padding and do not perform masking. The drawback of this method is that it does introduce additional computational costs compared to the cyclic shift. Fortunately, image dehazing networks tend to process much larger images than image patches at training

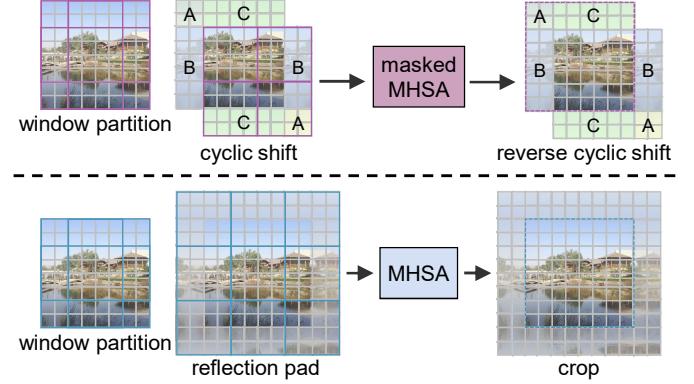


Fig. 5. Comparison of our proposed reflection padding scheme with cyclic shift scheme for shifted window partitioning. The actual percentage of the edge area is much smaller than the illustration.

time. When the image size becomes larger, the percentage of edge regions will become smaller.

#### E. W-MHSA with Parallel Convolution

We consider that multiplying MHSA is a low-pass filtering, a similar conclusion was presented in a very recent work [29]. Although the spatial information aggregation weight of MHSA is dynamic, the weight is always positive, making it work like smoothing. As a counterpart to MHSA's spatial information aggregation style, we perform additional convolution on  $V$ . Thus the spatial information aggregation scheme is

$$\begin{aligned} \text{Aggregation}(Q, K, V) = & \text{Softmax}\left(QK^T / \sqrt{d} + B\right)V \\ & + \text{Conv}(\hat{V}), \end{aligned} \quad (9)$$

where  $\hat{V} \in \mathbb{R}^{b \times h \times w \times c}$  denotes the  $V$  before window partitioning, and  $\text{Conv}(\cdot)$  can be either DWConv or a ConvBlock (Conv-ReLU-Conv). In other words, we still use the attention mechanism to aggregate information within the window, but also use convolution to aggregate information in the neighborhood without considering window partitioning. Furthermore, we discard the MHSA in some blocks, especially in the encoder's shallow stages and the decoder, and the revised block is shown in Fig. 2. The components illustrated with dashed boxes in the DehazeFormer block indicate they are optional. Specifically, some blocks do not contain MHSA and RescaleNorm, and reflection padding and cropping are used only when shifted window partitioning is required.

Note that a similar idea was proposed in CSwin Transformer [38], but we use the convolution to extract high frequency information instead of acting as a positional embedding. In contrast to CSwin Transformer, we use reflection padding instead of zero padding because we do not need it to encode position information implicitly [71]. Most importantly, DehazeFormer's convolutional layer is performed on  $\hat{V}$  before window partitioning, thus it provides the capability to aggregate information between windows.

#### F. Implementation Details

We provide five DehazeFormer's variants (-T, -S, -B, -M, and -L for tiny, small, basic, middle, and large, respectively).

TABLE I  
DETAILED ARCHITECTURE SPECIFICATIONS.

	Num. of Blocks	Embedding Dims	MLP Ratio	Attention Ratio	Num. of Heads	Conv Type
DehazeFormer-T	[ 4, 4, 4, 2, 2]	[24, 48, 96, 48, 24]	[2, 4, 4, 2, 2]	[1/4, 1/2, 3/4, 0, 0]	[2, 4, 6, 1, 1]	DWConv
DehazeFormer-S	[ 8, 8, 8, 4, 4]	[24, 48, 96, 48, 24]	[2, 4, 4, 2, 2]	[1/4, 1/2, 3/4, 0, 0]	[2, 4, 6, 1, 1]	DWConv
DehazeFormer-B	[16, 16, 16, 8, 8]	[24, 48, 96, 48, 24]	[2, 4, 4, 2, 2]	[1/4, 1/2, 3/4, 0, 0]	[2, 4, 6, 1, 1]	DWConv
DehazeFormer-M	[12, 12, 12, 6, 6]	[24, 48, 96, 48, 24]	[2, 4, 4, 2, 2]	[1/4, 1/2, 3/4, 0, 0]	[2, 4, 6, 1, 1]	ConvBlock
DehazeFormer-L	[16, 16, 16, 12, 12]	[48, 96, 192, 96, 48]	[2, 4, 4, 2, 2]	[1/4, 1/2, 3/4, 0, 0]	[2, 4, 6, 1, 1]	ConvBlock

TABLE I lists the detailed configurations of these variants. The attention ratio here indicates the percentage of blocks containing MHSA, and we place the blocks containing MHSA at the end of each stage. For the three small models (-T, -S, -B), we use DWConv ( $K = 5$ ) as the parallel convolutions. Because DWConv is an operation with low computational cost but high memory access cost [72], we use ConvBlock ( $K = 3$ ) for two large models (-M, -L).

When training, images are randomly cropped to  $256 \times 256$  patches. We set different mini-batch sizes for training different variants, *i.e.*,  $\{32, 16, 16, 16, 8\}$  for  $\{-T, -S, -B, -M, -L\}$ . Referring to the linear scaling rule [73], we set the initial learning rate to  $\{4, 2, 2, 2, 1\} \times 10^{-4}$  for  $\{-T, -S, -B, -M, -L\}$ . We use AdamW optimizer [74] with the cosine annealing strategy [75] to train the models, where the learning rate gradually decreases from the initial learning rate to  $\{4, 2, 2, 2, 1\} \times 10^{-6}$ .

#### IV. RS-HAZE DATASET

The RESIDE dataset is a large-scale homogeneous image dehazing dataset that advances the image dehazing. However, evaluating the method's capability of non-homogeneous image dehazing still relies on some small, unrealistic datasets [46], which use a haze machine to generate the non-homogeneous haze that would hardly exist. In contrast, remote sensing image dehazing is a practical non-homogeneous image dehazing task because the haze in remote sensing images is highly non-homogeneous. Therefore, we propose a new synthetic remote sensing image dehazing dataset named RS-Haze. Comparing to some remote sensing image dehazing datasets [55, 56, 76, 77], our proposed dataset is more realistic and larger scale.

##### A. Haze Synthesis Formulation

For generating remote sensing hazy images, researchers generally set  $d(x)$  to  $d_0$  since the remote sensing imaging system has a fixed imaging distance. However,  $d(x)$  is not the imaging distance but the thickness of the medium that scatters the light. Further, the haze medium in remote sensing images is non-homogeneous, making  $d(x)$  vary spatially but consistent over all channels. Besides, the transmission map  $t(x)$  is correlated with wavelength and haze conditions. Inspired by prior works [1, 2], we model the scattering coefficient as

$$\beta(\lambda, \gamma(x)) = c_0 \lambda^{-\gamma(x)}, \quad (10)$$

where  $c_0$  is a constant,  $\lambda$  is the channel's center wavelength, and the exponent  $\gamma(x)$  corresponds to the region-wise haze conditions. Combining Eq.(2) and Eq.(10), we can derive

$$t(x) = e^{-\beta(\lambda, \gamma(x))d(x)}. \quad (11)$$

Then the relationship of the transmission map between channel  $i$  and channel  $j$  can be expressed as

$$\ln t_i(x) / \ln t_j(x) = \beta_i(\lambda_i, \gamma(x)) / \beta_j(\lambda_j, \gamma(x)), \quad (12)$$

where  $t_{\{i,j\}}(x)$ ,  $\beta_{\{i,j\}}$ ,  $\lambda_{\{i,j\}}$  are the transmission map, scattering coefficient and center wavelength of channel  $\{i, j\}$ , respectively. If we take channel 1 as the reference channel, and the transmission map  $t_j(x)$  can be obtained via

$$t_j(x) = t_1(x)^{\left(\frac{\lambda_1}{\lambda_j}\right)^{\gamma(x)}}, \quad (13)$$

The final haze imaging model can be formulated as

$$I_j(x) = J(x)t_j(x) + A_j(1 - t_j(x)). \quad (14)$$

Here we can collect clean images  $J$  and set  $\lambda_j$  to the center wavelength of the corresponding channel. Thus the problem lies in how to obtain  $t_1(x)$ ,  $A_j$  and  $\gamma(x)$ .

##### B. Synthesis Pipeline

We first consider how to extract the transmission map  $t_1(x)$  from the real hazy images. The reflectance of the cirrus channel (channel 9) can characterize the spatially non-homogeneous properties of the natural haze [54], so we use it to generate the transmission map  $t_1(x)$  as

$$t_1(x) = 1 - \omega \rho_9(x), \quad (15)$$

where  $\rho_9(x)$  is the reflectance of the cirrus channel of the real hazy image, and  $\omega$  is a hyperparameter corresponding to the haze density. We find a large dark level in the cirrus channel, making the pixels over 5000 even in the haze-free image. Thus we apply a linear stretch of 0.1% to the cirrus channel to remove the dark level. If we do not remove the dark level, then the maximum value of  $t_1(x)$  is always smaller than 1, which is equivalent to an additional homogenous haze.

After that, we need to estimate the atmospheric light of the scene from the haze-free images. To this end, we regard the mean value of each channel's brightest 0.01% pixels as the atmospheric light [54]. However, there are still many cases of inaccurate estimation. Since the atmospheric light of each channel is correlated with each other, an additional constraint can be introduced to correct for the incorrectly estimated atmospheric light. Assume that the mean value of the estimated atmospheric light of all remote sensing images in channel  $i$  is  $\bar{A}_i$ . We set the reference values  $\bar{A}_r = (\bar{A}_6 + \bar{A}_7)/2$  and  $A_r = (A_6 + A_7)/2$ , and obtain the corrected atmospheric light of channel  $i$  by  $A'_i = A_r \cdot \bar{A}_i / \bar{A}_r$ . Fig. 6 shows how the correction refines the atmospheric light.

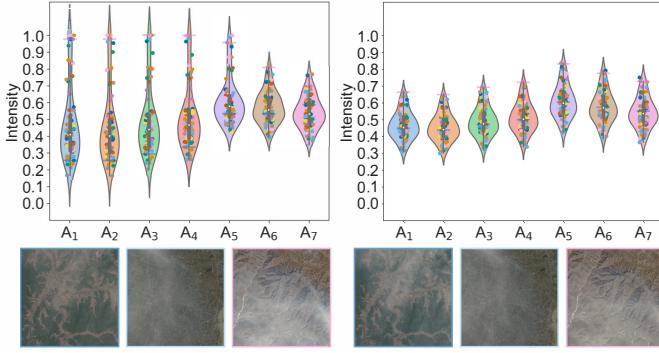


Fig. 6. Correction of the estimated atmospheric light. The top is atmospheric lights before and after correction, and the bottom is three synthesis samples.

TABLE II  
ATMOSPHERIC RELATIVE SCATTERING MODELS

Reflectance $\rho$	Condition	Exponent $\gamma$
$0.000 < \rho \leq 0.215$	very clear	4.0
$0.215 < \rho \leq 0.294$	clear	2.0
$0.294 < \rho \leq 0.373$	moderate	1.0
$0.373 < \rho \leq 0.451$	hazy	0.7
$0.451 < \rho \leq 0.529$	very hazy	0.5
$0.529 < \rho < 1.000$	cloudy	0.0

Finally, we need to obtain the  $\gamma(x)$ . Because the particle properties of haze can vary depending on the haze density [78], the exponent  $\gamma(x)$  should be modeled as a function of haze density. As shown in the Table II, we model the exponent  $\gamma(x)$  related to the haze reflectance  $\rho$ . We use  $\omega\rho_9(x)$  as the haze reflectance and fit the relationship between  $\gamma(x)$  and  $\omega\rho_9(x)$  with a cubic curve, which can be formulated as

$$\gamma(x) = a_3(\omega\rho_9(x))^3 + a_2(\omega\rho_9(x))^2 + a_1(\omega\rho_9(x)) + a_0, \quad (16)$$

where  $a_0 = 6.537$ ,  $a_1 = -27.465$ ,  $a_2 = 41.224$ , and  $a_3 = -21.547$ . Note that we clip  $\gamma(x)$  to  $[0, 4]$  to avoid outliers.

Now we can use Eq.(14) to synthesize the dataset. However, we found that the network trained with this dataset works well in the dense haze region of the synthetic image, but performs poorly on the dense haze region of the real image. We consider that, when the haze is dense, it is likely to block all the light from the ground [55]. According to the haze imaging model Eq.(14), even when  $t_j(x)$  is small and the synthesized haze is dense, there still exists information residuals from haze-free channel  $J_j(x)$ . To this end, we revise Eq.(14) as

$$I_j(x) = J_j(x)t_j(x)' + A_j(1 - t_j(x)), \quad (17)$$

where  $t_j(x)' = 1 - \xi(1 - t_j(x))$ , and we also clip  $t_j(x)'$  to  $[0, 1]$  to avoid outliers. Here  $t_j(x)$  is consistent with Eq.(13), but we introduce a decay factor  $\xi = 1.25$  to attenuate the information of  $J_j(x)$ . When the haze reaches a certain concentration, the synthesized hazy image completely lose the information of the hazy-free image in that region.

### C. Dataset Details

We download the multi-spectral (MS) images from the Landsat-8 Level 1 data product on EarthExplorer. We selected

TABLE III  
SUMMARY OF RS-HAZE DATASET

Name	Density	$\omega$	Train	Test
RS-Haze-L	Light	0.100-0.399	17100	900
RS-Haze-M	Moderate	0.400-0.699	17100	900
RS-Haze-D	Dense	0.700-0.999	17100	900
RS-Haze-mix	All	0.100-0.999	51300	2700

76 remote sensing images containing diverse topography with good weather conditions and performed atmospheric correction using the FLAASH module [79]. Meanwhile, 108 cloudy remote sensing images are selected to generate transmission maps using their cirrus channels. We crop  $512 \times 512$  image patches from the original remote sensing image using the GDAL library [80]. Finally, we obtained 6000 patches of haze-free MS images containing various terrain and 1500 patches of cirrus channels with a distribution similar to natural haze. Each haze-free image generates nine synthetic hazed images containing three different haze densities. The haze density is controlled by setting the range of  $\omega$ . The values of  $\omega$  in each range are obtained by sampling from the truncated Gaussian function. The summary of RS-Haze is shown in Table III.

## V. EXPERIMENTS

### A. Experimental setup

Our experiments are performed on the RESIDE dataset [52] and our RS-Haze dataset. The RESIDE dataset is one of the most commonly used datasets for image dehazing, and it contains three versions: RESIDE-V0, RESIDE-Standard, and RESIDE- $\beta$ . It contains several subsets, of which the most commonly used are: indoor training set (ITS), outdoor training set (OTS), and synthetic objective testing set (SOTS). We found that the existing works use different experimental setups and can be divided into two main categories: training on a combination of the ITS and the OTS and testing on the SOTS; training on the ITS and the OTS separately and testing on the indoor and outdoor scenes of the SOTS separately. For proving the effectiveness of DehazeFormer, we perform experiments on both setups, which we name RESIDE-Full and RESIDE-6K, respectively. We do not train large models under each experimental setup since the small models are good enough.

1) *RESIDE-Full*: Models are trained and tested on the indoor and outdoor scenes separately. Following FFA-Net [18], we use the full ITS (13,990 image pairs from RESIDE-Standard) and OTS (313,950 image pairs from RESIDE-V0) to train indoor models and outdoor models and test them on indoor scenes (500 image pairs) and outdoor scenes (500 image pairs) of the SOTS, respectively. In this experimental setup, all models are trained using their original training strategies, and we replicate the best results reported in the previous works. We train DehazeFormers on ITS for 300 epochs and on OTS for 30 epochs. Note that a few images in the outdoor subset are smaller than the configured patch size, so we discard these images during training. Besides, because the upper part of the outdoor image is often sky, we use only horizontal flipping for data augmentation.

TABLE IV  
QUANTITATIVE COMPARISON OF VARIOUS DEHAZING METHODS TRAINED ON THE RESIDE DATASETS.

Methods	ITS		OTS		RESIDE-6K		RS-Haze		Overhead	
	SOTS-indoor		SOTS-outdoor		SOTS-mix		RS-Haze-mix			
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	#Param	MACs
(TPAMI'10) DCP [4]	16.62	0.818	19.13	0.815	17.88	0.816	17.86	0.734	-	-
(TIP'16) DehazeNet [8]	19.82	0.821	24.75	0.927	21.02	0.870	23.16	0.816	0.009M	0.581G
(ECCV'16) MSCNN [9]	19.84	0.833	22.06	0.908	20.31	0.863	22.80	0.823	0.008M	0.525G
(ICCV'17) AOD-Net [11]	20.51	0.816	24.14	0.920	20.27	0.855	24.90	0.830	<u>0.002M</u>	<u>0.115G</u>
(CVPR'18) GFN [12]	22.30	0.880	21.55	0.844	23.52	0.905	29.24	0.910	0.499M	14.94G
(WACV'19) GCANet [14]	30.23	0.980	-	-	25.09	0.923	34.41	0.949	0.702M	18.41G
(ICCV'19) GridDehazeNet [13]	32.16	0.984	30.86	0.982	25.86	0.944	36.40	0.960	0.956M	21.49G
(CVPR'20) MSBDN [17]	33.67	0.985	33.48	0.982	28.56	0.966	38.57	0.965	31.35M	41.54G
(ECCV'20) PFDN [15]	32.68	0.976	-	-	28.15	0.962	36.04	0.955	11.27M	50.46G
(AAAI'20) FFA-Net [18]	36.39	0.989	<u>33.57</u>	<u>0.984</u>	<u>29.96</u>	<u>0.973</u>	<u>39.39</u>	<u>0.969</u>	4.456M	287.8G
(CVPR'21) AEGR-Net [19]	<u>37.17</u>	<u>0.990</u>	-	-	28.52	0.964	35.69	0.959	2.611M	52.20G
(ours) DehazeFormer-T	35.15	0.989	<b>33.71</b>	0.982	<b>30.36</b>	0.973	39.11	0.968	0.686M	6.658G
(ours) DehazeFormer-S	36.82	<b>0.992</b>	<b>34.36</b>	0.983	<b>30.62</b>	<b>0.976</b>	<b>39.57</b>	<b>0.970</b>	1.283M	13.13G
(ours) DehazeFormer-B	<b>37.84</b>	<b>0.994</b>	<b>34.95</b>	0.984	<b>31.45</b>	<b>0.980</b>	<b>39.87</b>	<b>0.971</b>	2.514M	25.79G
(ours) DehazeFormer-M	<b>38.46</b>	<b>0.994</b>	<b>34.29</b>	0.983	<b>30.89</b>	<b>0.977</b>	<b>39.71</b>	<b>0.971</b>	4.634M	48.64G
(ours) DehazeFormer-L	<b>40.05</b>	<b>0.996</b>	-	-	-	-	-	-	25.44M	279.7G

2) *RESIDE-6K*: Models are trained and tested on the mixed dataset. We use an experimental setup from DA [18], which differs significantly from the RESIDE-Full. Its training set contains 3,000 ITS image pairs and 3,000 OTS image pairs, and all images are resized to  $400 \times 400$ . Its testing set mixes indoor and outdoor image pairs to form a test set of 1,000 image pairs without resizing, here called SOTS-mix. In this experimental setup, we retrain all models using  $L_1$  loss on the RESIDE-6K training set for 1,000 epochs, and the learning rate is adjusted according to the model's mini-batch size. For some methods that estimate  $t(x)$ , we adapt them to predict the output image. Thus we can compare the architectures' performance, regardless of the impact of the training strategy.

3) *RS-Haze*: Models are trained on the RS-Haze-mix. For the default experimental setup, we use 8-bit gamma-corrected RGB images for training and testing. We train all models using  $L_1$  loss for 150 epochs, and other settings are the same as RESIDE-6K. For MS image dehazing, we use 16-bit linear images for training and testing. It aims to analyze the properties of MS and RGB images for image dehazing. Note that we compute PSNR and SSIM on the gamma-corrected RGB images when testing.

4) *Overhead*: We use the number of parameters (#Param) and multiply-accumulate operations (MACs) to measure the overhead. MACs are measured on  $256 \times 256$  images.

### B. Quantitative Comparison

We quantitatively compare the performance of DehazeFormers and baselines, and the results are shown in TABLE IV. Here we underline the best results in baselines and bold the results where DehazeFormers exceed them. Overall, our proposed DehazeFormers outperformed these baselines. We argue that the RESIDE-Full indoor set mainly measures the model's capability to handle high-frequency information, and

the outdoor set mainly measures the convergence speed of the model. RESIDE-6K measures the stability of the model and the capability to extract low-frequency information. RS-Haze measures the network's capability to extract semantic features. Notably, DehazeFormer-B sometimes outperforms DehazeFormer-M, indicating that the attention mechanism is more critical than convolution in these experimental setups.

1) *RESIDE-Full*: Training on ITS and testing on SOTS indoor set should be the most widely used experimental setup. Comparing the baseline methods, FFA-Net and AEGR-Net are far superior to the previous or contemporaneous methods. The former mainly relies on the large network, and the latter may also benefit from the proposed contrastive loss function. However, our proposed DehazeFormer-B surpasses all baseline methods in terms of PSNR and SSIM. Further, the PSNR of DehazeFormer-L exceeds 40 dB. To the best of our knowledge, this is the first method with the PSNR exceeding 40 dB on the SOTS indoor set, dramatically surpassing previous work. Finally, all variants of DehazeFormer work well, and we believe it is a scalable method. Unfortunately, some baselines do not report results on SOTS outdoor set. Because the training set of outdoor scenes consists of more than 300,000 sample pairs, DehazeFormers and baselines may not have converged. We believe that there is still much scope to improve the performance on SOTS outdoor set, and the current results reflect more the network's convergence speed. In particular, DehazeFormer-M is inferior to DehazeFormer-S on the outdoor set, probably because more nonlinear activation functions slow down the training. We remind that the results of baselines on RESIDE-Full are replicated from previous works, and some of them can achieve higher performance using our codebase.

2) *RESIDE-6K*: We found that the performance of all CNN-based networks under the RESIDE-6K experimental setup is worse than that of DehazeFormers, which we sup-

TABLE V  
PSNR / SSIM OF DEHAZEFORMER-S ON THE RGB / MS SET.

Setting	RS-Haze-L	RS-Haze-M	RS-Haze-D	RS-Haze-mix
RGB	43.68/0.993	39.58/0.979	35.46/0.938	39.57/0.970
MS	55.44/0.999	50.75/0.997	43.66/0.984	49.95/0.993

pose is due to the different image resolutions of the testing and training sets. Because the images of the training set are resized, its high-frequency information distribution is not consistent with the images of the testing set. As we argued, the convolutional layer is good at filtering high-frequency information, while the attention mechanism is good at filtering low-frequency information, making DehazeFormers perform better. We believe this property of the attention mechanism is important for image dehazing because it is not practical to collect dehazing datasets for each resolution setting.

3) *RS-Haze*: Compared with other experimental setups, the methods have higher PSNR on RS-Haze but lower SSIM. The scenes of remote sensing images are more monotonous than natural scenes. Thus, it is easier for methods to estimate images' latent color and brightness, making the PSNR higher. In contrast, the haze of RS-Haze is highly non-homogeneous, making the high-frequency information of the image corrupted and the SSIM accordingly lower. We compare the image dehazing methods on RS-Haze. It can be seen that FFA-Net is the best method in baselines, while our small model surpasses it. It is not only due to the excellent design of DehazeFormer itself but also because the remote sensing images have more similar regions, which are more favorable for self-attention [81]. Furthermore, the comparison of DehazeFormer-S on RGB and MS images is shown in TABLE V. As expected, dense haze is more difficult to be removed than light haze. Besides, the additional information provided by more channels and larger bit depths does improve the performance of the method substantially.

### C. Qualitative Comparison

We also select some samples of the results to analyze the performance of each method qualitatively. Since we do not retrain the baselines on RESIDE-Full, we only show the test results on RESIDE-6K and RS-Haze. Fig. 7 and Fig. 8 illustrate qualitative comparisons of our DehazeFormer-S with some representative learning-based dehazing methods.

1) *RESIDE-6K*: We select four samples taken from different scenes in the SOTS mix set to evaluate the network's dehazing performance, including synthetic indoor and outdoor haze. AOD-Net and GCANet produce severe color distortions, which make their indoor and outdoor results too dim or too bright. Though color is restored in most areas of images dehazed by PFDN and FFA-Net, color distortion remains on distant objects and small objects near the edge of images. By comparison, the color is recovered correctly through the haze by our DehazeFormer-S, and the results look natural and realistic. When it comes to the region where haze density varies significantly in some indoor scenes, as shown in the

enlarged white boxes in the second row in Fig. 7, we can observe that almost all the comparative methods fail to remove the haze effectively. However, our DehazeFormer-S restores clear images well, keeps texture and color information, and contains the least haze residual.

2) *RS-Haze*: Three images taken from different scenes with different haze densities in the RS-Haze are selected to evaluate the network's dehazing performance on non-homogeneous haze. AOD-Net can barely handle non-homogeneous haze and produces severe artifacts. GCANet, PFDN, and FFA-Net can remove haze effectively when the haze is thin, as shown in the first two rows in Fig. 8, but they are not as good as DehazeFormer-S in color and detail reproduction. Moreover, DehazeFormer-S can remove dense haze, while all other networks produce apparent artifacts. See the water surface area in the third row of Fig. 8.

### D. Ablation Study

We perform ablation studies on the RESIDE-Full's indoor scene. However, because not every DehazeFormer block has an MHSA, these blocks degenerate into meaningless linear layers when removing the parallel convolution. So we build DehazeFormer-A for ablation studies only. In particular, we set the attention ratio of DehazeFormer-A to 1 and reduce the depth of the network to keep the computational cost and parameters. TABLE VI lists the difference between DehazeFormer-T and DehazeFormer-A. We can see that DehazeFormer-T has better performance than DehazeFormer-A. In terms of overhead, DehazeFormer-A has fewer parameters but a higher computational cost compared to DehazeFormer-T. Note that we find that the results of ablation studies on different datasets are not always consistent, *e.g.*, RESIDE-6K prefers DehazeFormer with a high attention ratio compared to the RESIDE-Full indoor set. We mark the results in red if there is an improvement compared to the baseline (DehazeFormer-A) and in blue if there is a degradation.

1) *Normalization layer*: We study normalization layers and their placements on the performance, and the results are shown in TABLE VII. We can see that avoiding the loss of inter-patch relativity and reintroducing the lost statistics does improve the networks' performance. Besides, the normalization layer is more critical for MHSA than MLP. Considering that the normalization layer before MLP has no significant impact on the performance, removing it makes sense since it is not cheap to obtain the standard deviation of the feature maps. However, the negative impact of LayerNorm is not as evident as expected since the normalization layer showed a severe impact on performance in our early ablation studies on RESIDE-6K. Thus we plan to explore the relationship between the dataset and the normalization layer in our future work.

2) *Shifted window partitioning scheme*: We study the schemes of shifted window partitioning, and the results are shown in TABLE VIII. Because masked padding and masked cyclic shift are equivalent in terms of spatial information aggregation, we train only a single network. If we replace the reflection padding with zero padding, the network's performance drops significantly. Zero-padding introduces meaningless tokens, and the attention matrix is all-positive, making



Fig. 7. Qualitative comparison of image dehazing methods on SOTS mix set, where the first two rows are indoor images, and the last two rows are the outdoor images. The first column is the hazy images and the last column is the corresponding ground truth.

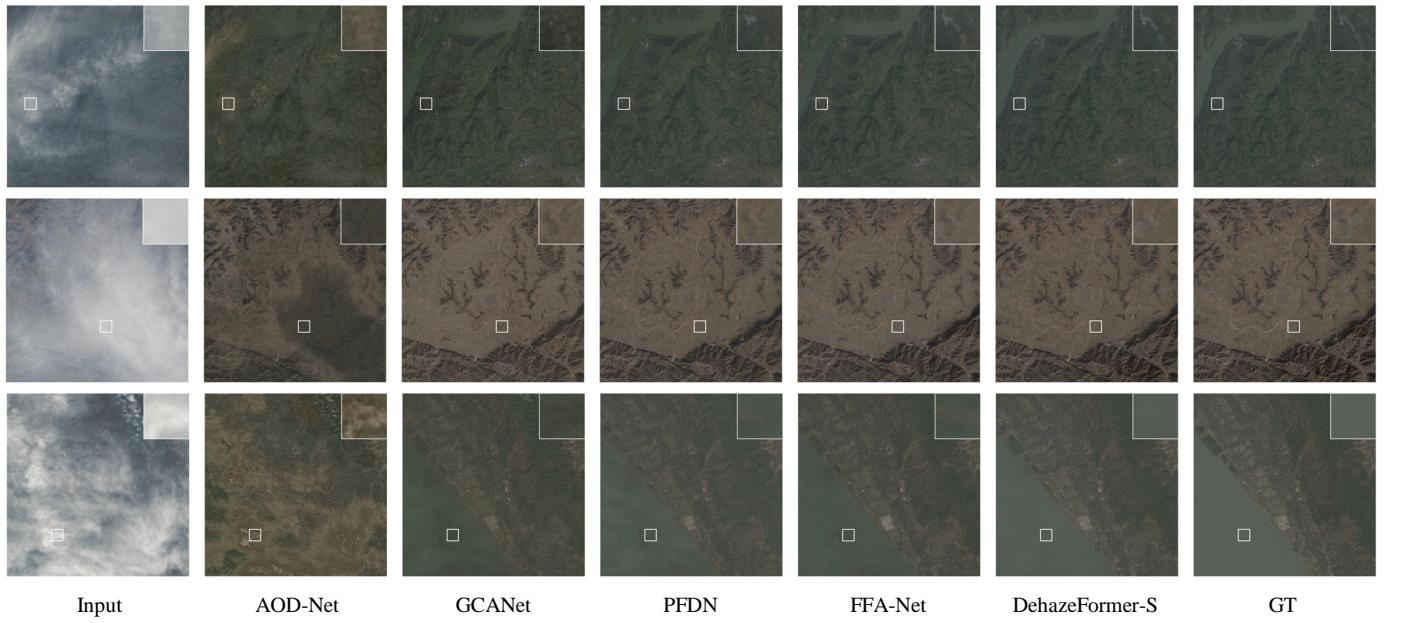


Fig. 8. Qualitative comparison of image dehazing methods on RS-Haze. The first column is the hazy images and the last column is the ground truth.

useless information mixed in. In contrast, cyclic shift without mask also introduces unreasonable interactions between tokens but has a less negative impact. Finally, our proposed scheme gives a moderate performance improvement to the network. Considering that it only introduces a negligible additional computational cost on  $256 \times 256$  images, it is worthwhile.

*3) Nonlinear activation functions:* We study the difference in the nonlinear activation functions, and the results are shown in TABLE IX. We replace all nonlinear activation functions in the network, including the nonlinear activation functions in MLP and SK fusion layer. Surprisingly, the nonlinear acti-

vation functions dramatically affect the network performance, while our early ablation studies on RESIDE-6K did not show such a huge gap. The networks using ReLU and LeakyReLU perform about the same because they are both piecewise linear functions that can be easily inverted. Although the form of SoftReLU is simple, it is not easily inverted, so the networks with it yield significant performance degradation. Furthermore, GELU is non-monotonic, and it is more difficult to be inverted, making the networks with it perform very poorly. We argue that it is essential to consider the invertibility of the nonlinear activation function when building the network.

TABLE VI  
COMPARISON BETWEEN DEHAZEFORMER-T AND DEHAZEFORMER-A.

	Num. of Blocks	MLP Ratio	Attention Ratio	Num. of Heads	PSNR	SSIM	#Param	MACs
DehazeFormer-T	[4, 4, 4, 2, 2]	[2, 4, 4, 2, 2]	[1/4, 1/2, 3/4, 0, 0]	[2, 4, 6, 1, 1]	35.15	0.989	0.686M	6.658G
DehazeFormer-A	[2, 2, 2, 2, 2]	[2, 4, 4, 4, 2]	[1, 1, 1, 1, 1]	[2, 4, 6, 4, 2]	34.85	0.988	0.671M	7.185G

TABLE VII  
ABLATION STUDY ON NORMALIZATION LAYERS.

Setting	PSNR	SSIM	#Param	MACs
DehazeFormer-A	34.85	0.988	0.671M	7.185G
RescaleNorm → LayerNorm <sup>†</sup>	34.73	0.988	0.668M	7.175G
→ LayerNorm	34.45	0.987	0.668M	7.175G
– PreNorm for MHSA	34.17	0.986	0.668M	7.163G
+ PreNorm for MLP	34.86	0.987	0.674M	7.207G

TABLE IX  
ABLATION STUDY ON NONLINEAR ACTIVATION FUNCTIONS.

Setting	PSNR	SSIM	#Param	MACs
DehazeFormer-A	34.85	0.988	0.671M	7.185G
ReLU → GELU	33.02	0.983	0.671M	7.279G
→ SoftReLU (0.1)	33.73	0.985	0.671M	7.229G
→ LeakyReLU (0.1)	34.89	0.988	0.671M	7.229G

TABLE X  
ABLATION STUDY ON PARALLEL CONV LAYERS.

Setting	PSNR	SSIM	#Param	MACs
DehazeFormer-A	34.85	0.988	0.671M	7.185G
→ Parallel DWConv	32.90	0.983	0.653M	6.910G
→ Parallel DWConv on $X$	33.99	0.987	0.671M	7.185G
→ DWConv before MLP	33.49	0.986	0.671M	7.185G

TABLE XI  
ABLATION STUDY ON OTHER COMPONENTS.

Setting	PSNR	SSIM	#Param	MACs
DehazeFormer-A	34.85	0.988	0.671M	7.185G
Soft Recon. → Recon.	34.50	0.987	0.671M	7.171G
SK Fusion → Cat Fusion	34.78	0.988	0.673M	7.256G

## VI. CONCLUSION

This paper introduces various improvements for Swin Transformer applied to image dehazing, and the proposed DehazeFormer achieves superior performance on several datasets. To summarize, we propose to use RescaleNorm and ReLU to replace the commonly used LayerNorm and GELU to avoid some negative effects that are not important for high-level vision tasks but critical for low-level vision tasks. To improve the capability of MHSA, we propose a shifted window partitioning scheme based on reflection padding and a spatial information aggregation scheme using convolution in parallel with attention. We also propose some minor improvements that are applicable to other networks. Finally, we collect a large-scale remote sensing image dehazing dataset to evaluate the network's capability to remove highly non-homogeneous haze, and DehazeFormer also achieves an impressive performance. In the future, we plan to work on more lightweight and more straightforward architectures and extend the architecture to other low-level vision tasks. Besides, encoding feature maps on thumbnails and then decoding them on the original image may achieve real-time 4K image dehazing.

4) *Parallel conv*: We study to prove the importance of parallel convolution with attention: a) remove the parallel convolution; b) place the convolution parallel with the MHSA, i.e., the input to the convolution is  $X$  instead of  $V$ ; c) place the convolution before the MLP [69], and the results are shown in TABLE X. As can be seen, additional convolutional layers in the Transformer block can dramatically improve the network's performance, but their placement is critical. Inserting DWConv into the FFN brings only minor performance, although the scheme has been widely employed in previous work. We consider that the transformer works somehow because it separates intra-token and inter-tokens interactions into two steps, while inserting DWConv in FFN would break this property. DWConv in parallel with attention is better than DWConv in parallel with MHSA. Although both schemes use DWConv to aggregate spatial information, the former is done in the same feature space as attention, while the latter is done in a different feature space. DWConv provides static learnable aggregation weights, while attention provides dynamic all-positive aggregation weights. Thus the convolution parallel to attention does play a complementary role with attention.

5) *Other components*: We verify the impact of the soft reconstruction module and SK fusion module on the network's performance. Although SK fusion brings only a minor performance gain, we consider it a good alternative to concatenation fusion, given its lower overhead. Whereas the soft reconstruction brings more improvement than expected, we believe introducing soft constraints on prior is beneficial.

## REFERENCES

- [1] Earl J McCartney. Optics of the atmosphere: scattering by molecules and particles. *New York*, 1976. 1, 6
- [2] Shree K Nayar and Srinivasa G Narasimhan. Vision in bad weather. In *ICCV*, volume 2, pages 820–827. IEEE, 1999. 6
- [3] Srinivasa G Narasimhan and Shree K Nayar. Vision and the atmosphere. *IJCV*, 48(3):233–254, 2002. 1
- [4] Kaiming He, Jian Sun, and Xiaou Tang. Single image haze removal using dark channel prior. *IEEE TPAMI*, 33(12):2341–2353, 2010. 1, 2, 8
- [5] Raanan Fattal. Dehazing using color-lines. *ACM TOG*, 34(1):1–14, 2014. 2
- [6] Qingsong Zhu, Jiaming Mai, and Ling Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE TIP*, 24(11):3522–3533, 2015. 2
- [7] Dana Berman, Shai Avidan, et al. Non-local image dehazing. In *CVPR*, pages 1674–1682, 2016. 1, 2
- [8] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE TIP*, 25(11):5187–5198, 2016. 1, 2, 8
- [9] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks. In *ECCV*, pages 154–169, 2016. 2, 8
- [10] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *CVPR*, pages 3194–3203, 2018. 1, 2
- [11] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *ICCV*, pages 4770–4778, 2017. 2, 8
- [12] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *CVPR*, pages 3253–3261, 2018. 2, 8
- [13] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Grid-dehazenet: Attention-based multi-scale network for image dehazing. In *ICCV*, pages 7314–7323, 2019. 1, 2, 8
- [14] Dongdong Chen, Mingming He, Qingnan Fan, Jing Liao, Liheng Zhang, Dongdong Hou, Lu Yuan, and Gang Hua. Gated context aggregation network for image dehazing and deraining. In *WACV*, pages 1375–1383, 2019. 2, 8
- [15] Jiangxin Dong and Jinshan Pan. Physics-based feature dehazing networks. In *ECCV*, pages 188–204. Springer, 2020. 8
- [16] Qili Deng, Ziling Huang, Chung-Chi Tsai, and Chia-Wen Lin. Hardgan: A haze-aware representation distillation gan for single image dehazing. In *ECCV*, pages 722–738. Springer, 2020.
- [17] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale boosted dehazing network with dense feature fusion. In *CVPR*, pages 2157–2167, 2020. 8
- [18] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. Ffa-net: Feature fusion attention network for single image dehazing. In *AAAI*, pages 11908–11915, 2020. 2, 7, 8
- [19] Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Contrastive learning for compact single image dehazing. In *CVPR*, pages 10551–10560, 2021. 2, 8
- [20] Chao Wang, Hao-Zhen Shen, Fan Fan, Ming-Wen Shao, Chuan-Sheng Yang, Jian-Cheng Luo, and Liang-Jian Deng. Eaa-net: A novel edge assisted attention network for single image dehazing. *KBS*, 228:107279, 2021. 1, 2
- [21] Yuanjie Shao, Lerenhan Li, Wenqi Ren, Changxin Gao, and Nong Sang. Domain adaptation for image dehazing. In *CPVR*, pages 2808–2817, 2020. 1
- [22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017. 1, 2
- [23] Wenhui Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 1, 2
- [24] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, pages 558–567, 2021. 2
- [25] Krzysztof Choromanski, Valerii Likhoshevstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *ICLR*, 2021. 3
- [26] Tete Xiao, Piotr Dollar, Mannat Singh, Eric Mintun, Trevor Darrell, and Ross Girshick. Early convolutions help transformers see better. In *NeurIPS*, volume 34, 2021. 3
- [27] Zihang Dai, Hanxiao Liu, Quoc Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In *NeurIPS*, volume 34, 2021.
- [28] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. In *ICLR*, 2022.
- [29] Namuk Park and Songkuk Kim. How do vision transformers work? In *ICLR*, 2022. 2, 3, 5
- [30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 1, 3
- [31] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021. 3
- [32] Qihang Yu, Yingda Xia, Yutong Bai, Yongyi Lu, Alan L Yuille, and Wei Shen. Glance-and-gaze vision transformer. In *NeurIPS*, volume 34, 2021.
- [33] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. In *ICLR*, 2022.
- [34] Hezheng Lin, Xing Cheng, Xiangyu Wu, Fan Yang, Dong Shen, Zhongyuan Wang, Qing Song, and Wei Yuan. Cat: Cross attention in vision transformer. *arXiv preprint arXiv:2106.05786*, 2021. 3
- [35] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. *arXiv preprint arXiv:2106.02689*, 2021. 3
- [36] 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.
- [37] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS*, volume 34, 2021. 3
- [38] 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 *CVPR*, 2022. 3, 5
- [39] Sitong Wu, Tianyi Wu, Haoru Tan, and Guodong Guo. Pale transformer: A general vision transformer backbone with pale-shaped attention. *arXiv preprint arXiv:2112.14000*, 2021.
- [40] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. In *NeurIPS*, volume 34, 2021. 1, 3
- [41] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2, 4
- [42] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 2
- [43] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 2, 4
- [44] Vinod Nair and Geoffrey E Hinton. Rectified linear units

- improve restricted boltzmann machines. In *ICML*, 2010. 2, 4
- [45] Xiang Li, Wenhui Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, pages 510–519, 2019. 2, 3
- [46] Codruta O Ancuti, Cosmin Ancuti, and Radu Timofte. Nhaze: An image dehazing benchmark with non-homogeneous hazy and haze-free images. In *CVPR Workshop*, pages 444–445, 2020. 2, 6
- [47] Cosmin Ancuti, Codruta O Ancuti, Radu Timofte, and Christophe De Vleeschouwer. I-haze: a dehazing benchmark with real hazy and haze-free indoor images. In *ACIVS*, pages 620–631. Springer, 2018.
- [48] Codruta O Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In *ICCV Workshop*, pages 754–762, 2018.
- [49] Codruta O Ancuti, Cosmin Ancuti, Mateu Sbert, and Radu Timofte. Dense-haze: A benchmark for image dehazing with dense-haze and haze-free images. In *ICIP*, pages 1014–1018. IEEE, 2019. 2
- [50] Cosmin Ancuti, Codruta O Ancuti, and Christophe De Vleeschouwer. D-hazy: A dataset to evaluate quantitatively dehazing algorithms. In *ICIP*, pages 2226–2230. IEEE, 2016. 2
- [51] Yanfu Zhang, Li Ding, and Gaurav Sharma. Hazerd: an outdoor scene dataset and benchmark for single image dehazing. In *ICIP*, pages 3205–3209, 2017.
- [52] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE TIP*, 28(1):492–505, 2018. 2, 7
- [53] Manjun Qin, Fengying Xie, Wei Li, Zhenwei Shi, and Haopeng Zhang. Dehazing for multispectral remote sensing images based on a convolutional neural network with the residual architecture. *IEEE J-STARS*, 11(5):1645–1655, 2018. 2
- [54] Jianhua Guo, Jingyu Yang, Huanjing Yue, Hai Tan, Chunping Hou, and Kun Li. Rsdehazenet: Dehazing network with channel refinement for multispectral remote sensing images. *IEEE TGRS*, 59(3):2535–2549, 2020. 6
- [55] Binghui Huang, Li Zhi, Chao Yang, Fuchun Sun, and Yixu Song. Single satellite optical imagery dehazing using sar image prior based on conditional generative adversarial networks. In *WACV*, pages 1806–1813, 2020. 6, 7
- [56] Aditya Mehta, Harsh Sinha, Murari Mandal, and Pratik Narang. Domain-aware unsupervised hyperspectral reconstruction for aerial image dehazing. In *WACV*, pages 413–422, 2021. 2, 6
- [57] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2
- [58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 3
- [59] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV Workshop*, pages 1833–1844, 2021. 3, 4
- [60] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. In *CVPR*, 2022. 3, 4
- [61] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1251–1258, 2017. 3
- [62] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 3
- [63] Dong Zhao, Jia Li, Hongyu Li, and Long Xu. Hybrid local-global transformer for image dehazing. *arXiv preprint arXiv:2109.07100*, 2021. 3
- [64] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *CVPR*, 2022. 3
- [65] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 3
- [66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4
- [67] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 4
- [68] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR, 2019. 4
- [69] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 4, 11
- [70] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 4
- [71] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? In *ICLR*, 2020. 5
- [72] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, pages 116–131, 2018. 6
- [73] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 6
- [74] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6
- [75] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 6
- [76] Yufeng Li and Xiang Chen. A coarse-to-fine two-stage attentive network for haze removal of remote sensing images. *IEEE GRSL*, 18(10):1751–1755, 2020. 6
- [77] Faramarz Naderi Darbaghshahi, Mohammad Reza Mohammadi, and Mohsen Soryani. Cloud removal in remote sensing images using generative adversarial networks and sar-to-optical image translation. *IEEE TGRS*, 60:1–9, 2021. 6
- [78] Pat S Chavez Jr. An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data. *Remote sensing of environment*, 24(3):459–479, 1988. 7
- [79] T Cooley, Gail P Anderson, Gerald W Felde, Michael L Hoke, Anthony J Ratkowski, James H Chetwynd, James A Gardner, Steven M Adler-Golden, Michael W Matthew, Alexander Berk, et al. Flaash: a modtran4-based atmospheric correction algorithm, its application and validation. In *IGARSS*, volume 3, pages 1414–1418. IEEE, 2002. 7
- [80] Frank Warmerdam. The geospatial data abstraction library. In *Open source approaches in spatial data handling*, pages 87–104. Springer, 2008. 7
- [81] Yiqun Mei, Yuchen Fan, Yulun Zhang, Jiahui Yu, Yuqian Zhou, Ding Liu, Yun Fu, Thomas S Huang, and Honghui Shi. Pyramid attention networks for image restoration. *arXiv preprint arXiv:2004.13824*, 2020. 9