IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 September 2025, accepted 11 October 2025, date of publication 14 October 2025, date of current version 21 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3621348

RESEARCH ARTICLE

SWSformer: Subwindow Shuffle Transformer for Image Restoration

NAGAYUKI OKITSU AND MASATO SHIRAI
Graduate School of Natural Science and Technology, Shimane University, Matsue, Shimane 690-8504, Japan
Corresponding author: Masato Shirai (shirai@cis.shimane-u.ac.jp)

ABSTRACT Transformers have demonstrated superior performance over conventional methods in various tasks due to their ability to capture long-range dependencies and adaptively generate weights. However, their computational complexity increases quadratically with the number of tokens, limiting their applicability to high-resolution image tasks. To address this problem, recent image restoration methods have attempted to mitigate this limitation by adopting mechanisms that reduce computational demands at the expense of fully capturing long-range dependencies. In contrast, the window shuffle self-attention (WS-SA) mechanism of the Shuffle Transformer reduces computational complexity without significantly limiting the ability to capture long-range dependencies. Nevertheless, WS-SA suffers from generating excessively sparse self-attention maps with a limited receptive field when applied to high-resolution images. In this work, we propose subwindow shuffle self-attention (SWS-SA), a mechanism that expands the receptive field without increasing the computational complexity of WS-SA. SWS-SA introduces a subwindow-based spatial shuffle to enhance the receptive field. Additionally, we apply average pooling to the query embeddings to further reduce computational complexity. Furthermore, we present the Subwindow Shuffle Transformer (SWSformer), which employs SWS-SA as its core component and integrates effective techniques from related works. To evaluate the performance of SWSformer, we conduct experiments on image denoising, deblurring, and deraining tasks. The experimental results demonstrate that SWSformer achieves state-of-the-art performance. Moreover, we perform comprehensive ablation studies to identify the contributions of individual components and settings to the model's overall effectiveness.

INDEX TERMS Deep learning, attention, transformer, image restoration.

I. INTRODUCTION

The widespread use of camera-equipped devices, such as smartphones, has increased the frequency of image degradation caused by factors like noise, blurring, and rain. These degradations typically manifest irregularly across an image, making manual removal difficult. In addition, image restoration is considered an ill-posed problem because a single degraded image can correspond to multiple potential restored outputs. To address this ill-posed nature and achieve stable performance, deep learning-based image restoration methods have been actively developed in recent years.

Many deep learning-based image restoration methods are built on convolutional neural networks (CNNs) [1], [2], [3], [4], [5], [6], [7]. The fundamental operation in CNNs is convolution, which uses grid-based weights called kernels to extract features, ensuring local connectivity and translation equivariance. Although these characteristics enable CNNs to perform stable image restoration, their receptive fields are limited, thus making them incapable of capturing long-range dependencies. Since this limitation constrains restoration performance, developing image restoration techniques that can model long-range dependencies is crucial for achieving superior results.

The attention mechanism of the Transformer [8], [9] is highly effective for capturing long-range dependencies. This mechanism computes adaptive weights based on the

The associate editor coordinating the review of this manuscript and approving it for publication was Sudhakar Radhakrishnan.

relationships between all pixels and aggregates pixel features using these weights. However, the computational complexity of attention scales quadratically with input resolution, making its application to high-resolution image restoration challenging. To address this, recent state-of-the-art (SOTA) image restoration methods have adopted more efficient attention mechanisms. Wang et al. proposed Uformer [10], which uses shifted window self-attention (W-SA) [11]. Zamir et al. proposed Restormer [12] and Chen et al. proposed NAFNet [13], both of which combine channel attention (CA) and convolution. Although these methods are highly efficient, their ability to capture long-range dependencies is limited because W-SA's receptive field is restricted to local windows and CA cannot capture spatial dependencies. To address these limitations, Ghasemabadi et al. proposed CGNet [14], which applies successive convolutions with varying kernel sizes. Although this method can gradually expand its receptive field, it risks missing important dependencies because it cannot directly capture long-range dependencies.

The failure of recent SOTA methods to model long-range dependencies causes noticeable artifacts in image restoration, most notably the loss of edge information. For instance, Fig. 1 shows that whereas the ground truth (GT) image contains a clear brown window in the upper right, its lower edge is largely lost in the images restored by these methods. This happens because the models cannot leverage the global context provided by long-range dependencies. We analyze this failure using Local Attribution Maps (LAM) [15], as shown in the lower part of Fig. 1. LAM, which uses integrated gradients to visualize contributing pixels, reveals that these SOTA methods rely almost exclusively on local information around the degraded edge. Because the edge in the noisy input is obscured, accurately restoring it requires capturing long-range dependencies to recognize the window as a complete structure.

In this paper, we propose subwindow shuffle self-attention (SWS-SA), a novel attention mechanism that captures long-range dependencies with minimal computational overhead. SWS-SA extends the attention mechanism from the Shuffle Transformer [16], which was initially designed to create connections between local windows. We adapt this foundation to significantly improve its capacity for modeling long-range dependencies. Based on our SWS-SA, we present the Subwindow Shuffle Transformer (SWSformer), a new architecture that integrates SWS-SA with effective techniques from related work. The first is the Positional Encoding Generator (PEG), proposed by Chu et al. [17], which encodes absolute positions using a 3×3 convolution. The second is the Neighbor Window Connection (NWC) module, proposed by Huang et al. [16], which addresses the window grid problem by using a convolution with larger kernels. The third is the Gated-Dconv Feed-Forward Network (GDFN), proposed by Zamir et al. [12], which enhances local dependencies through 3×3 convolutions equipped with a gating mechanism. Although these mechanisms are powerful individually, combining them yields further synergistic effects. First, our SWS-SA, by its design, has limitations in capturing detailed local information. Therefore, incorporating the convolution-based PEG, NWC, and GDFN can compensate for this weakness. Furthermore, since PEG/GDFN and NWC use different kernel sizes, their combination enables multi-scale feature extraction.

To demonstrate the superiority of our SWSformer, we evaluate our method on image denoising, deblurring, and deraining tasks, where it achieves state-of-the-art performance. In addition, we conduct comprehensive ablation studies to quantify the impact of each component and design choice on the model's overall performance.

The main contributions of this work are summarized as follows:

- We propose subwindow shuffle self-attention (SWS-SA), a computationally efficient attention mechanism designed to solve the key problem of capturing long-range dependencies in image restoration.
- We introduce the Subwindow Shuffle Transformer (SWSformer), a novel architecture built upon SWS-SA. SWSformer achieves SOTA performance in image denoising and competitive results in image deblurring and deraining.
- We perform comprehensive ablation studies to validate our design choices and quantify the impact of each component on SWSformer's performance.

II. RELATED WORKS

A. IMAGE RESTORATION

Image restoration aims to generate a clean image from a degraded one. Recently, deep learning-based methods have achieved SOTA performance, most of which are based on the U-Net [18] for multi-scale feature extraction via its hierarchical structure. Furthermore, convolution is widely used for its properties of local connectivity and translation equivariance. Among convolution-based methods, Zhang et al. proposed DBLRNet [6] for video deblurring using 3D convolutions, Huang et al. proposed GraNet [7], which detects various rain streaks, and Chen et al. proposed NAFNet [13], which is composed of a simple, nonlinearity-free architecture. Although these approaches offer an excellent trade-off between efficiency and performance, their restoration capabilities are limited because convolution cannot capture long-range dependencies. To address this, methods incorporating the attention mechanism of the Transformer [8] have recently been proposed, and several surveys [19], [20] have demonstrated their effectiveness. However, the high computational complexity of standard attention makes its application to image restoration challenging. Consequently, recent methods employ efficient attention mechanisms. Wang et al. proposed Uformer [10] using shifted W-SA [11], and Zamir et al. proposed Restormer [12] using CA. Although these methods are highly efficient, they are limited in capturing long-range dependencies because

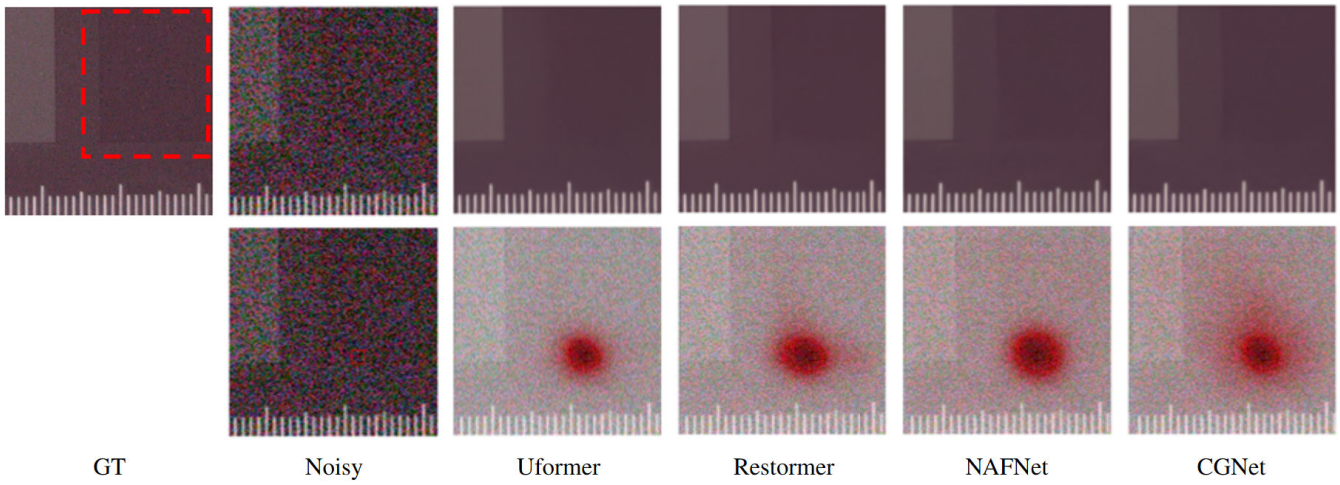


FIGURE 1. Upper row: Denoising results of recent state-of-the-art (SOTA) methods. Lower row: Analysis results using Local Attribution Maps (LAM). The red highlighted areas in the LAM results indicate the pixels used to denoise the region within the red box of the noisy image. As shown by the LAM results, recent SOTA methods rely only on local information near the lower edge of the brown window. This reliance on local information leads to the failure to restore the edge.

W-SA's receptive field is confined to a local window, and CA is incapable of modeling spatial dependencies. To overcome this, Ghasemabadi et al. proposed CGNet [14] and Gao et al. proposed FADformer [21], both of which indirectly capture long-range dependencies by convolution. However, this indirectness risks overlooking important dependencies. In this paper, we propose a novel attention mechanism that directly captures long-range spatial dependencies while keeping the computational cost low.

B. VISION TRANSFORMERS

Originally developed for natural language processing [8], [22], [23], Transformers have recently been adapted for computer vision tasks, including image recognition [9], [24], [25], segmentation [26], [27], [28], and object detection [11], [29], [30]. The core of a Vision Transformer (ViT) is self-attention, a mechanism that computes relationships among all image patches to capture long-range dependencies and generate adaptive weights. This has led to remarkable performance across various tasks. The primary drawback, however, is that self-attention's computational complexity scales quadratically with the number of patches. This scaling presents a significant problem for high-resolution images. A common solution is window-based self-attention [11], [31], [32], [33], [34], which computes attention within local windows. Whereas this approach is efficient, it cannot capture global long-range dependencies because its receptive field is restricted. This limitation hinders the performance of ViT models. To solve this problem, we propose a novel window-based self-attention mechanism that maintains a global receptive field.

C. SHUFFLE TRANSFORMER

The Swin Transformer [11] uses a shift mechanism to establish connections between adjacent windows, which alleviates the problem of a purely local receptive field. However, this method is insufficient for building robust, long-range connections across windows. To solve this problem, Huang et al. proposed the Shuffle Transformer (Shuffle T). Shuffle T applies a spatial shuffle to the windows, grouping patches from the same relative position across different windows to establish long-range connections. Furthermore, it introduces a Neighbor Window Connection module between the self-attention and MLP layers. This module uses depthwise convolution to mitigate grid-like artifacts, a problem inherent in window-based shuffling. Inspired by Shuffle T's ability to efficiently capture long-range dependencies, our work adapts its core attention mechanism for image restoration tasks.

III. METHOD

The primary objective of this work is to develop an image restoration method capable of capturing long-range dependencies with minimal computational complexity. To achieve this, we introduce a novel attention mechanism called subwindow shuffle self-attention (SWS-SA), which utilizes subwindow-based spatial shuffling and average pooling. We then present the Subwindow Shuffle Transformer (SWSformer), a new architecture that incorporates SWS-SA with effective components from related studies. This section begins with an overview of the SWSformer architecture. Next, we detail its intra-block structure, with a particular focus on our proposed SWS-SA mechanism.

A. OVERALL PIPELINE

Recent deep learning-based image restoration methods employ U-Net [18], which consists of an encoder-decoder as an overall structure. In U-Net, the hierarchical structure extracts multi-scale features. In this work, as shown in Fig. 2, we adopt U-Net following the recent image restoration

methods. Given a degraded image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$, we first apply a 3×3 convolution to extract low-level features $\mathbf{F_0} \in \mathbb{R}^{C\times H\times W}$. Where C is the number of channels, and H and W are the height and width of the input. Then $\mathbf{F_0}$ passes through the encoder. In the encoder, the features are gradually reduced in spatial size and expanded in number of channels by downsampling. Next, a low-resolution feature $\mathbf{F_l} \in \mathbb{R}^{16C\times \frac{H}{16}\times \frac{W}{16}}$ passes through the middle block. The output of the middle block passes through a decoder. In the decoder, the features are gradually restored to the original size by upsampling, and the deep feature $\mathbf{F_d} \in \mathbb{R}^{C\times H\times W}$ is output. In addition, in the decoder, the output of the encoder is skip-connected. Finally, the residual image $\mathbf{R} \in \mathbb{R}^{3\times H\times W}$ is generated by 3×3 convolution, and then it is added to the degraded image to obtain the restored image $\hat{\mathbf{I}} = \mathbf{I} + \mathbf{R}$.

B. INTRA-BLOCK STRUCTURE

In our intra-block structure, we integrate our proposed SWS-SA with two mechanisms proposed in related works: the Neighbor Window Connection (NWC) module [16] and the Gated Dconv Feed-Forward Network (GDFN) [12]. First, NWC is composed of a depthwise convolution (dwconv) with a large kernel. Its primary role is to mitigate the grid artifacts caused by the window-based mechanism of SWS-SA. These artifacts, which appear as boundary lines between independently processed windows, can be suppressed by aggregating information across the window boundaries. NWC achieves this by applying a dwconv with a kernel size similar to the window size. Unlike standard convolution, which processes spatial and channel features simultaneously, dwconv applies an independent kernel to each channel to extract only spatial features, while also being significantly more computationally efficient. This allows NWC to focus on spatial aggregation, effectively suppressing grid artifacts. Furthermore, the high computational efficiency of dwconv offsets the increased computational cost associated with using a large kernel. We apply NWC after SWS-SA to prevent subsequent mechanisms from being affected by these artifacts. Second, GDFN is a mechanism that combines convolution with a gating mechanism. Its purpose is to further refine the output of SWS-SA by capturing local dependencies. In GDFN, a feature map is fed into two paths. In each path, a 1×1 convolution first expands the channels, followed by a 3×3 dwconv. The output of one path is passed through a GELU activation function [35] and then multiplied by the output of the other path. This gating mechanism allows the network to selectively emphasize important features. Finally, a 1×1 convolution aggregates the information across channels. The use of 3×3 dwconv allows GDFN to effectively extract spatial local dependencies. Moreover, the computational efficiency of dwconv is crucial for mitigating the computational overhead introduced by the gating mechanism. We apply GDFN after NWC to further refine the grid-artifact-free feature map with local dependency information and the gating mechanism. Finally, the intra-block structure is formulated as follows:

$$\begin{aligned} X_l &= \text{SWS-SA}(Z_{l-1}) + Z_{l-1} \\ Y_l &= \text{NWC}(X_l) + X_l \\ Z_l &= \text{GDFN}(Y_l) + Y_l \end{aligned} \tag{1}$$

where X_l, Y_l, Z_l are the output features of SWS-SA, NWC, and GDFN in the l-th block, respectively.

C. SUBWINDOW SHUFFLE SELF-ATTENTION

Capturing full long-range dependencies along the spatial dimension requires the standard self-attention (SA) used in Vision Transformers [9]. The primary drawback of this approach is its computational complexity, $\text{O}(H^2W^2C)$, which scales quadratically with the input resolution. This scaling presents a significant problem for high-resolution image restoration. A common solution is to adopt the window self-attention (W-SA) from the Swin Transformer [11], as done in several methods [10], [36]. Although this window-based approach is computationally efficient, it restricts the receptive field to local windows, thereby preventing the model from capturing global long-range dependencies.

To solve the aforementioned problem, we propose a novel attention mechanism named subwindow shuffle self-attention (SWS-SA), which utilizes a subwindow-based spatial shuffle and average pooling. Our approach builds upon the self-attention mechanism of the Shuffle Transformer (Shuffle T) [16], which we term "window shuffle self-attention" (WS-SA) for the purposes of this paper. The WS-SA process, illustrated in Fig. 3(a), begins by partitioning the input into nonoverlapping windows. It then applies a spatial shuffle, which groups pixels from the same relative position across all windows. This operation establishes connections among all windows, enabling the capture of long-range dependencies. The final step is to apply multihead self-attention (MHSA) to these shuffled windows. The computational complexity of the query-key dot-product in WS-SA for a window of size $M \times M$ is given by:

$$\text{O}(M^2HWC) = \frac{HW}{M^2} \times M^2 \times M^2 \times C \tag{2}$$

As indicated by Eqn. 2, the computational complexity of WS-SA scales linearly with input resolution, allowing it to efficiently capture long-range dependencies in high-resolution images. However, a significant problem arises when the shuffled window size M is much smaller than the input resolution HW: the self-attention becomes overly sparse (Fig. 4(a)). This sparsity can prevent the model from combining relevant features and can cause a loss of contextual consistency, which leads to unstable performance. Whereas one might consider increasing the window size M to address this, its quadratic impact on computational complexity (Eqn. 2) makes this approach impractical.

The core challenge, therefore, is to increase the effective window size without incurring a higher computational cost.

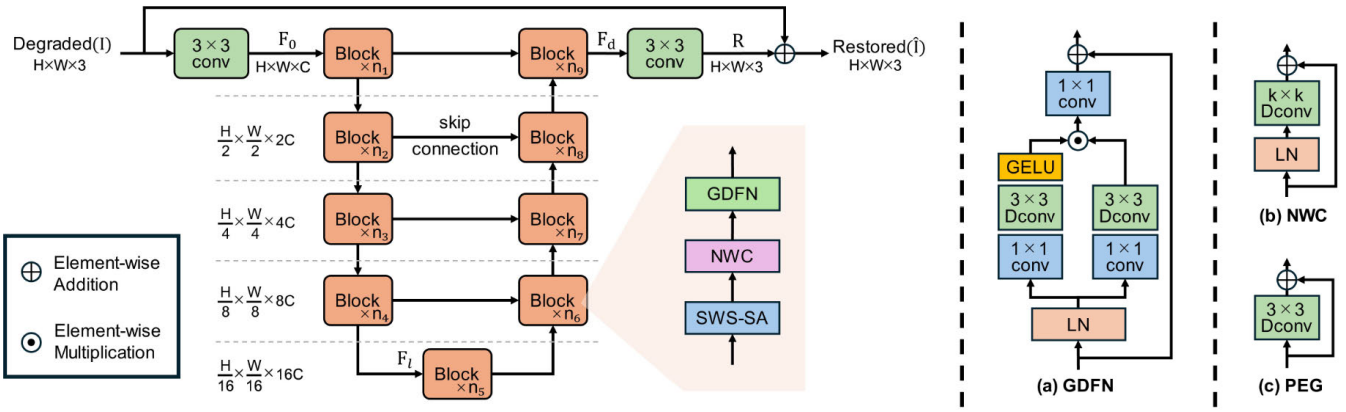


FIGURE 2. Overall pipeline of SWSformer. The SWSformer block consists of subwindow shuffle self-attention (SWS-SA), the Neighbor Window Connection (NWC) module [16], and the Gated-Dconv Feed-Forward Network (GDFN) [12]. LN denotes Layer Normalization [37] and Dconv denotes Depthwise convolution. The Position Encoding Generator (PEG) [17] is a positional encoding method incorporated into SWS-SA.

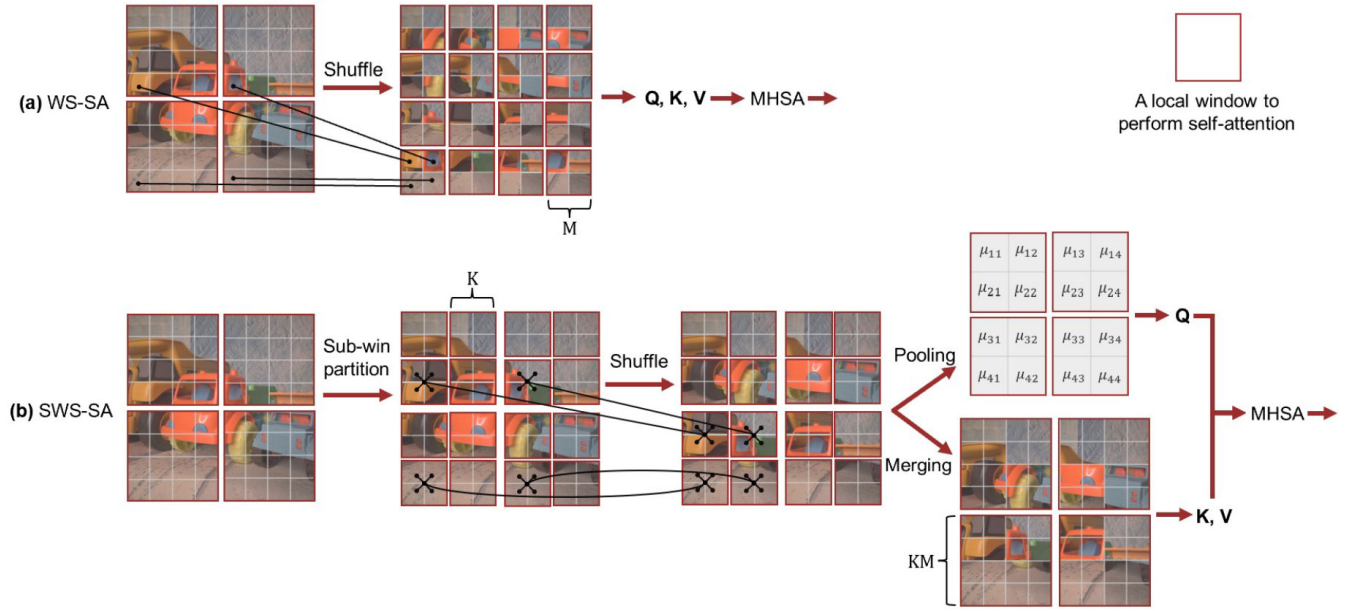


FIGURE 3. Details and comparison of the Shuffle Transformer's window shuffle self-attention (WS-SA) [16] and our subwindow shuffle self-attention (SWS-SA). Q, K, V are the query, key, and value, respectively; MHSA denotes multihead self-attention; and μ_{ij} is the average value of the (i, j) subwindow. The spatial shuffle operation groups tokens or subwindows at the same position in all windows.

Our proposed SWS-SA solves this problem by using subwindows and average pooling. As illustrated in Fig. 3(b), SWS-SA first partitions each window into $K \times K$ subwindows. It then applies a spatial shuffle on a subwindow basis, which groups subwindows from the same relative position across all windows. This process expands the effective shuffled window size to $KM \times KM$. To keep the computation manageable, average pooling is applied only to the queries within each subwindow. Consequently, the computational complexity of the query-key dot-product for SWS-SA is given by:

$$\mathrm{O}(M^2HWC) = \frac{HW}{K^2M^2} \times M^2 \times K^2M^2 \times C \qquad (3)$$

As shown in Eqn. 3, SWS-SA maintains the same computational complexity as WS-SA. This allows our method to expand the receptive field by a factor of K^2 at no additional computational cost, directly mitigating the sparsity problem (Fig. 4(b)). As described above, our SWS-SA captures long-range dependencies over a large receptive field by shuffling subwindows, while simultaneously reducing computational complexity by average pooling the queries, thus achieving a balance between efficiency and high-quality attention. The output of SWS-SA, which is downsampled by average pooling, must be restored to its original dimensions. This is accomplished first through nearest-neighbor interpolation, followed by an inverse process consisting of an inverse shuffle and window merging.

Positional information is crucial for self-attention to accurately assess feature relevance. Whereas the original

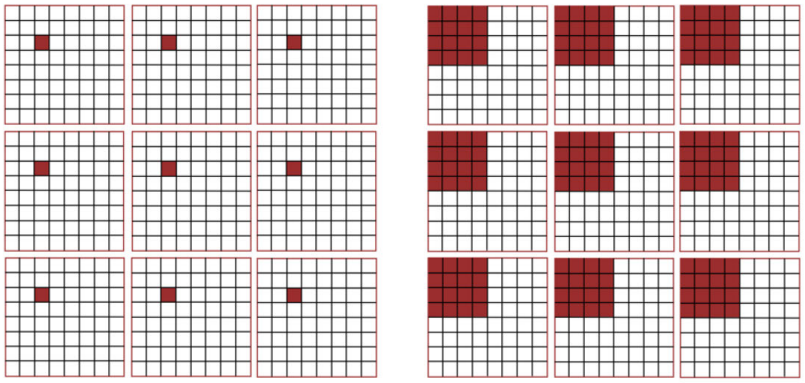


FIGURE 4. Comparison of receptive field size between the Shuffle Transformer's window shuffle self-attention (WS-SA) [16] and our subwindow shuffle self-attention (SWS-SA). Red areas indicate the receptive fields. Whereas the receptive field of WS-SA is very sparse, our SWS-SA achieves a denser field and effectively mitigates this sparsity.

Shuffle T uses relative position bias (RPB) [11], this approach is incompatible with our SWS-SA due to the use of average pooling on a subwindow basis, which obscures relative positional data. To solve this problem, we adopt the Position Encoding Generator (PEG) [17]. PEG implicitly encodes the absolute position using a 3×3 dwconv with zero-padding. The spatially-focused convolution by the depthwise operation facilitates the precise acquisition of positional information while minimizing the computational cost. Since position information needs to be encoded before the application of attention, we apply PEG before SWS-SA.

The PEG, NWC, and GDFN each have distinct roles and are indispensable for maximizing the performance of SWS-SA. Moreover, the combination of these mechanisms means that convolutions with various kernel sizes are applied within the block. This creates synergistic effects, such as compensating for SWS-SA's limitation in capturing detailed local dependencies and enabling multi-scale feature extraction. The complete formulation for SWS-SA is as follows:

$$
\begin{aligned}
X' &= \mathrm{Partition}(\mathrm{PEG}(\mathrm{LN}(X))),\\
X^{\mathrm{s}} &= \mathrm{Shuffle}(X'),\\
X^{\mathrm{s}}_{\mathrm{p}} &= \mathrm{AvgPool}(X^{\mathrm{s}}),\\
X^{\mathrm{s}}_{\mathrm{m}} &= \mathrm{Merge}(X^{\mathrm{s}}),\\
Y^{\mathrm{s}} &= \mathrm{Attention}(X^{\mathrm{s}}_{\mathrm{p}}W^{\mathrm{Q}}_i, X^{\mathrm{s}}_{\mathrm{m}}W^{\mathrm{K}}_i, X^{\mathrm{s}}_{\mathrm{m}}W^{\mathrm{V}}_i),\\
Y &= \mathrm{Inverse}(\mathrm{NNI}(Y^{\mathrm{s}})),
\end{aligned}
$$

$$
\mathrm{Attention}(Q, K, V) = \mathrm{SoftMax}(\frac{QK^{\mathrm{T}}}{\sqrt{d}})V \tag{4}
$$

where $X \in \mathbb{R}^{C\times H\times W}$ are the input features, LN is the Layer Normalization [37], $X', X^{\mathrm{s}} \in \mathbb{R}^{M^2\times K^2\times C}$ are the features partitioned into subwindow and features after spatial shuffle respectively, $X^{\mathrm{s}}_{\mathrm{p}} \in \mathbb{R}^{M^2\times C}$ are output features of the average pooling, $X^{\mathrm{s}}_{\mathrm{m}} \in \mathbb{R}^{M^2K^2\times C}$ are subwindow merged features, Q, K, V are query, key, and value, d is the query/key dimension, $W^{\mathrm{Q}}, W^{\mathrm{K}}, W^{\mathrm{V}}$ are the projection matrices of query, key and value, $Y^s \in \mathbb{R}^{M^2\times C}$ are the output features of the attention, $Y \in \mathbb{R}^{C\times H\times W}$ are the features expanded by nearest-neighbor interpolation (NNI) and restored to the original shape by the inverse process.

IV. EXPERIMENTS

To evaluate the performance of our SWSformer, we conduct experiments on image denoising, deblurring, deraining, low-light enhancement, and multi-weather nighttime image restoration tasks. Additionally, we perform comprehensive ablation studies to analyze the contributions of individual components and settings to the model's overall performance.

A. EXPERIMENTAL SETUP

To compare performance, we train our SWSformer and several recent image restoration models. The models other than SWSformer are prepared using their respective official GitHub repositories [38], [39], [40], [41], [42], [43], [44], [45]. For this experiment, we train SWSformer for 40 epochs with the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.9$, and a weight decay of 0) and the PSNR loss function. The initial learning rate is set to $1e^{-3}$ and gradually reduced to $1e^{-7}$ using a cosine annealing schedule. On the other hand, we train all models other than SWSformer for 40 epochs with their original configurations. Note that we apply our training configuration to Restormer, as it achieves better performance than with its original settings. For all models, we set the batch size to 6 and the training patch size to 256×256. We train each model three times with different random seeds and report the average scores.

We measure performance using two standard metrics, the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM), which are formulated as

follows:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}_\mathrm{I}^2}{\mathrm{MSE}}\right)$$

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{5}$$

Here, x and y are the model output and the ground-truth image, respectively. MAX_I is the maximum possible pixel value of image I, and MSE is the mean squared error. The terms μ_x and σ_x represent the mean and variance of x, whereas σ_{xy} is the covariance between x and y. The stabilization variables are $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, where L is the dynamic range of the pixel values, and the constants $k_1, k_2 \ll 1$ are set to 0.01 and 0.03.

In addition, to evaluate model efficiency, we measure the multiply-accumulate operations (MACs), number of parameters, and latency for each model.

1) DENOISING SETUP

For the denoising experiment, we use the SIDD dataset [46], a real-world dataset comprising images taken by five smartphone cameras under 10 different lighting conditions. This dataset consists of 30,608 training images and 1,280 test images. We train the models for a total of 204,053 iterations (i.e., 40 epochs). In this experiment, we compare our SWSformer with recent image restoration methods: Restormer [12], NAFNet [13], and CGNet [14].

2) DEBLURRING SETUP

In the deblurring experiment, we use the GoPro dataset [47], which consists of blurry images taken by a GoPro and corresponding sharp ground-truth images from a high-speed camera. This dataset contains 16,824 training images and 1,111 test images. We train the models for a total of 112,160 iterations (i.e., 40 epochs). In this experiment, we compare our SWSformer with recent image restoration methods: Restormer [12], NAFNet [13], and CGNet [14].

In addition, we use the Test-time Local Converter (TLC) [48] and a multihead mechanism [49], in line with recent image restoration methods [13], [14]. In TLC, global information handling mechanisms such as SWS-SA are applied to local patches during inference to suppress performance degradation caused by differences in patch size between training and inference. The multihead mechanism can efficiently improve performance by dividing the channels of the model's output.

3) DERAINING SETUP

In the deraining experiment, we use the Rain100H dataset [50], which is composed of synthetic rainy images. This dataset originally consisted of 1,800 training images and 100 test images. However, a subsequent study [51] revealed that approximately 500 training images shared the same backgrounds as the test set. Therefore, we use the 1,255 training images from which they have been removed. We train the models for a total of 8,400 iterations (i.e., 40 epochs) and compute PSNR and SSIM scores using the Y channel in YCbCr color space, following previous works [12], [21]. In this experiment, we compare our SWSformer with the recent image restoration methods NAFNet [13] and CGNet [14], and the deraining model FADformer [21].

4) LOW-LIGHT ENHANCEMENT SETUP

In the low-light enhancement experiment, we use the LOLv2 dataset [52], which is a real-world dataset consisting of paired low-light and corresponding well-illuminated images. This dataset consists of 689 training images and 100 test images. We train the models for a total of 4,600 iterations (i.e., 40 epochs). In this experiment, we compare our SWSformer with the recent image restoration methods NAFNet [13] and CGNet [14], and the low-light enhancement model DarkIR [53]. We train DarkIR using the official model implementation [45] in our environment, as its training code has not been released.

5) MULTI-WEATHER NIGHTTIME IMAGE RESTORATION SETUP

In the multi-weather nighttime image restoration experiment, we use the AllWeatherNight dataset [54], which is a large-scale dataset designed for restoring nighttime images affected by multiple adverse weather effects and flares. This dataset is comprised of multiple datasets, and covers various single and combined degradations, including haze, rain streaks, raindrops, and snow. In this experiment, we use the multi-degradation rain scene and snow scene datasets. These datasets are composed of 3,000 and 2,500 training images and 300 and 200 test images, respectively. We train the models for a total of 20,000 iterations on the rain scene and 16,680 iterations on the snow scene. We compare our SWSformer with the recent image restoration methods NAFNet [13] and CGNet [14].

6) MODEL CONFIGURATION

The configuration of our SWSformer is based on NAFNet [13]. The model consists of a 5-level encoder-decoder. The number of encoder blocks is $(2, 2, 4, 8)$, the number of middle blocks is 12, and the number of decoder blocks is $(2, 2, 2, 2)$. We set the model width to 64 and the kernel sizes of the NWC module to $(7, 5, 3, 3, 3)$. For the SWS-SA, the shuffled window size is 16, the subwindow size is 8, and the number of heads is $(1, 2, 4, 8, 16)$. Since the smaller model size achieved higher performance on the low-light enhancement task and the multi-weather nighttime image restoration task, we set the model width to 32, the kernel sizes of the NWC module to $(3, 3, 3, 3, 3)$, and the shuffled window size and subwindow size to 8 for these two tasks.

The configurations for the models other than SWSformer follow the default settings of their respective papers.

B. RESULTS

In the results tables, the best score is **highlighted** and the second best score is underlined.

1) DENOISING RESULTS

Table 1 shows a comparison of the denoising results. Our SWSformer achieves a 0.1dB gain over the second-best method, CGNet. Regarding efficiency, SWSformer requires fewer MACs and reduces latency by approximately half compared to Restormer, another Transformer model. Nevertheless, its number of parameters is the largest of the four models, which is due to the SWSformer block's design, incorporating multiple mechanisms like SWS-SA, NWC, and GDFN.

2) DEBLURRING RESULTS

Table 2 shows a comparison of the deblurring results. Our SWSformer achieves a 0.41dB gain over the second-best method, the multihead version of NAFNet. Regarding efficiency, the results follow the same trend as the denoising experiments: our SWSformer is more efficient than the Transformer-based Restormer, while also having the largest number of parameters among the four models. We present the visual restoration results in Fig. 5, where our SWSformer restores more fine-grained details than the other methods.

3) DERAINING RESULTS

Table 3 shows a comparison of the deraining results. Our SWSformer achieves a 0.11dB gain over the second-best method, CGNet. In terms of efficiency, SWSformer has the highest computational cost among the four models. This is because the other models, being composed of convolutions, are specifically designed for efficiency. We present the visual restoration results in Fig. 6, where our SWSformer removes rain streaks most effectively.

4) LOW-LIGHT ENHANCEMENT RESULTS

In this experiment, we observed that the grid artifacts appeared in the SWSformer's restoration results. To mitigate this, we apply an overlapping window partition to the SWS-SA during inference. Table 4 shows a comparison of the low-light enhancement results. Our SWSformer-S achieves a 1.34dB gain over the second-best method, CGNet while keeping the computational cost low. We present the visual restoration results in Fig. 7, where our SWSformer restores clear images with low noise.

5) MULTI-WEATHER NIGHTTIME IMAGE RESTORATION RESULTS

For the same reason as in the low-light enhancement experiment, we apply an overlapping window partition to the SWS-SA during inference. Table 5 shows a comparison of the multi-weather nighttime image restoration results. While keeping the computational cost low, our SWSformer-S achieves significant gains over the second-best methods. Specifically, it outperforms NAFNet by 1.44dB on rain scene dataset and CGNet by 3.73dB on snow scene dataset. We present the visual restoration results in Fig. 8, where our SWSformer generates the clearest images.

6) ANALYSIS OF PERFORMANCE SCALING

To analyze how each model's performance scales with the training period, we illustrate the denoising performance every 10 epochs in Fig. 9. As shown in Fig. 9, our SWSformer consistently achieves the highest performance at all training stages. We can also observe that SWSformer's performance scales in a similar manner to recent SOTA methods like CGNet and NAFNet. This implies that the performance gap among these models remains stable throughout the training period, indicating that our SWSformer can maintain its superior performance even at a high epoch count.

C. DENOISING RESULTS WITH LARGE NUMBER OF EPOCHS

In the above experiments, we use a small number of epochs compared to recent image restoration methods to reduce experimental costs. Methods like NAFNet [13] and CGNet [14] train models for 400K iterations with a batch size of 64, which corresponds to approximately 836 epochs on the SIDD dataset. Therefore, to demonstrate that SWSformer outperforms other methods with more extensive training, we train it on the SIDD dataset for 836 epochs. For this experiment, we set the batch size to 24 and train SWSformer for a total of 1,066,666 iterations.

The results are shown in Table 6. In this table, the PSNR and SSIM values for methods other than SWSformer are taken from their respective papers. As shown in the table, our SWSformer achieves a 0.25dB gain over the previous SOTA method, CGNet. Furthermore, we present the visual restoration results in Fig. 10. In the first image (previously mentioned in the introduction), our SWSformer captures long-range dependencies to clearly restore the edge of the brown window. In the second image, SWSformer restores an object that other methods fail to reconstruct. In the third image, SWSformer clearly restores even fine objects such as letters. In the fourth and fifth images, SWSformer restores the fine details most cleanly.

D. ABLATION STUDIES

We conduct ablation studies to analyze the contributions of the various components and settings to SWSformer's high performance. The primary difference between SWSformer and other recent methods is our novel attention mechanism, subwindow shuffle self-attention (SWS-SA). Therefore, our analysis begins with SWS-SA, investigating its optimal configuration and its performance benefits over the original Shuffle Transformer.

Additionally, we investigate the effects of several other key components:

TABLE 1. Comparison of denoising performance on SIDD dataset.

Method	PSNR ↑	SSIM ↑	MACs (G) ↓	Params (M) ↓	Latency (ms) ↓
Restormer [38]	39.86	0.959	141.0	26.11	2295.89
NAFNet [39]	39.86	0.959	63.2	115.98	612.14
CGNet [40]	39.92	0.959	61.7	119.22	722.31
SWSformer (Ours)	**40.02**	**0.960**	132.8	194.98	1435.65

TABLE 2. Comparison of deblurring performance on GoPro dataset. local: Test-time Local Converter [48], MH: multihead mechanism [49].

Method	PSNR ↑	SSIM ↑	MACs (G) ↓	Params (M) ↓	Latency (ms) ↓
Restormer-local [41]	30.31	0.934	141.0	26.11	2295.89
NAFNet-MH-local [42]	31.32	0.945	63.4	67.89	455.13
CGNet-MH-local [40]	31.30	0.945	61.8	68.77	481.48
SWSformer-MH-local (Ours)	**31.73**	**0.950**	133.1	194.98	1391.67

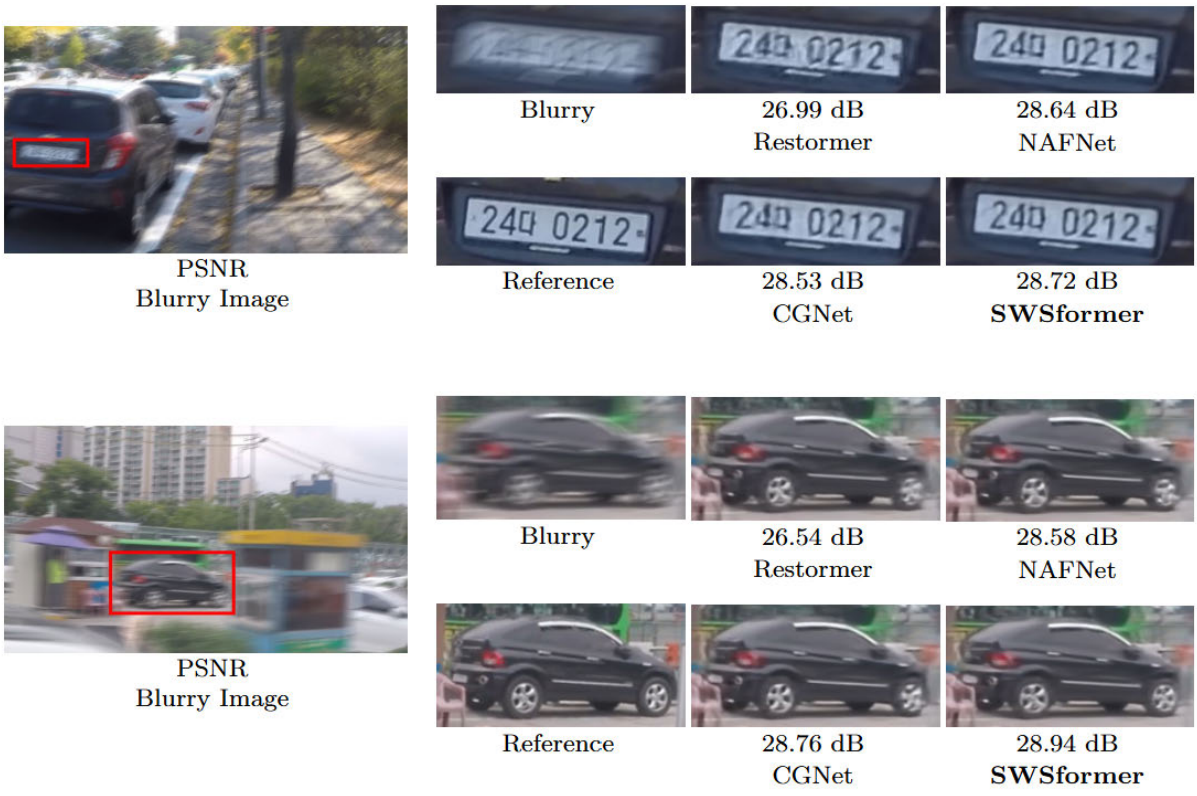


FIGURE 5. Deblurring results on GoPro dataset.

TABLE 3. Comparison of deraining performance on Rain100H dataset.

Method	PSNR ↑	SSIM ↑	MACs (G) ↓	Params (M) ↓	Latency (ms) ↓
FADformer [44]	28.80	0.872	48.2	7.44	1165.31
NAFNet [39]	29.17	0.876	63.2	115.98	612.14
CGNet [40]	29.24	0.876	61.7	119.22	722.31
SWSformer (Ours)	**29.35**	**0.879**	132.8	194.98	1435.65

TABLE 4. Comparison of low-light enhancement performance on LOLv2 dataset.

Method	PSNR ↑	SSIM ↑	MACs (G) ↓	Params (M) ↓	Latency (ms) ↓
NAFNet [39]	18.94	0.797	63.2	115.98	612.14
CGNet [40]	19.15	0.793	61.7	119.22	722.31
DarkIR [45]	19.16	0.779	7.1	3.32	190.52
SWSformer-S (Ours)	**20.50**	**0.807**	29.7	49.13	571.69

- Neighbor Window Connection (NWC) module [16]: A standard Transformer block consists of attention and a feedforward network. We investigate the effect of SWSformer's unusual structure,

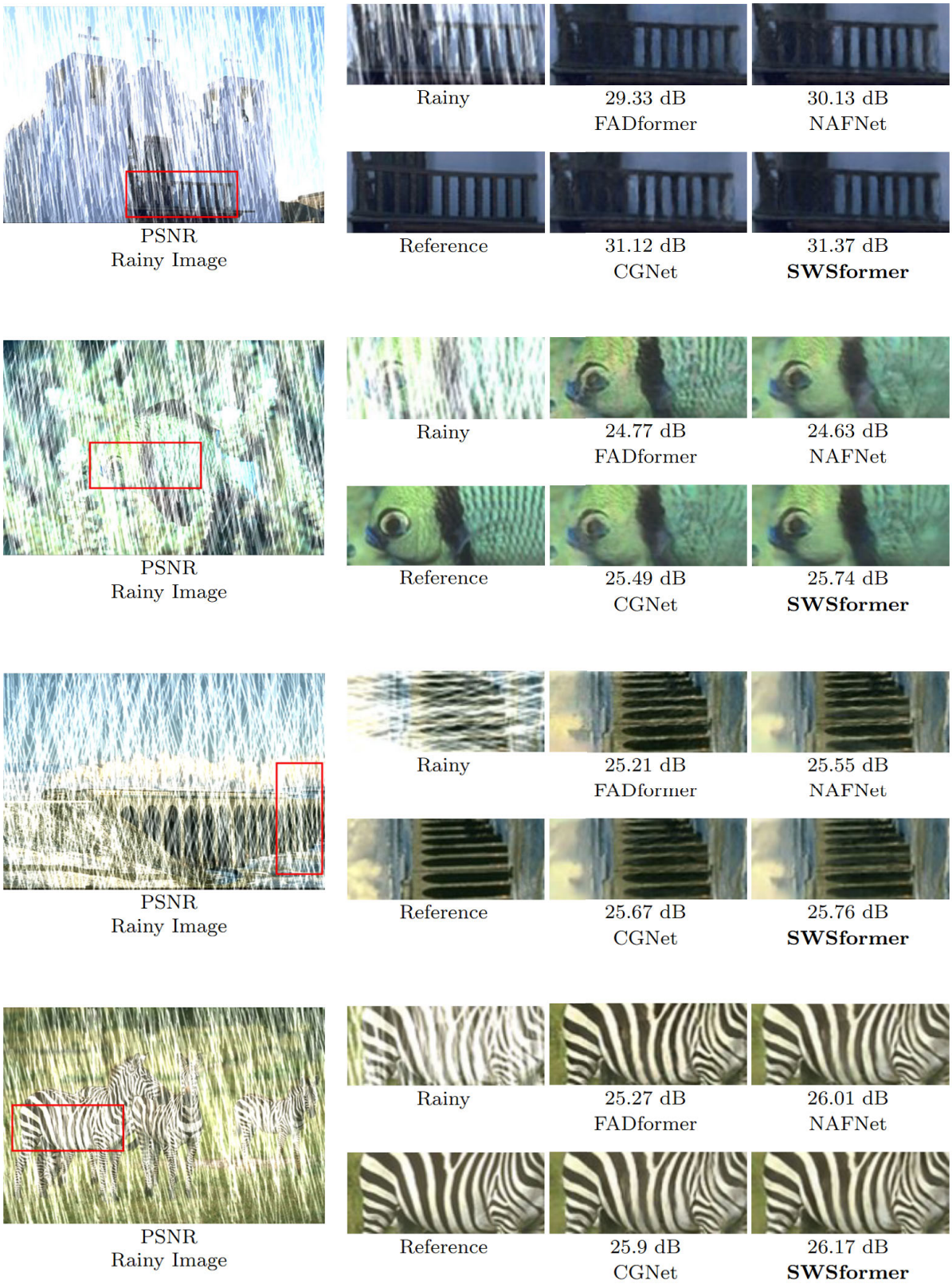


FIGURE 6. Deraining results on Rain100H dataset.

which includes the NWC module between these two layers.

- Positional Encoding: Because the query average pooling in SWS-SA can make it difficult to capture positional

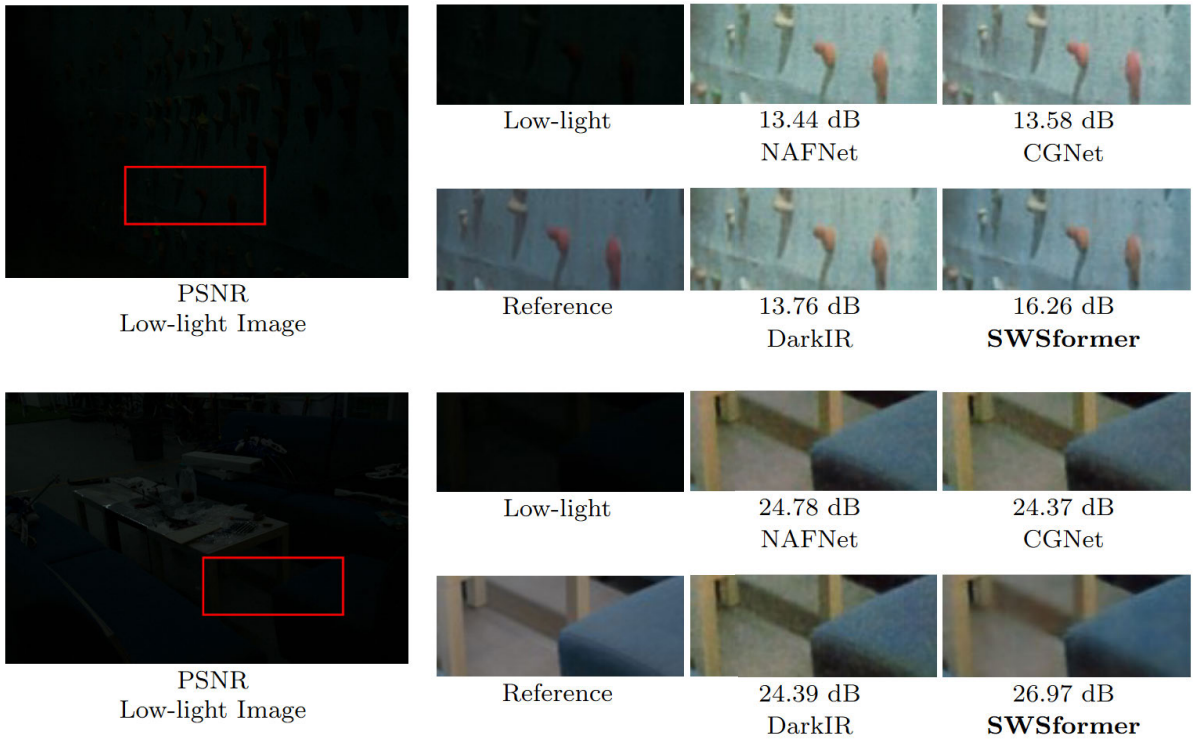


FIGURE 7. Low-light enhancement results on LOLv2 dataset.

TABLE 5. Comparison of multi-weather nighttime image restoration performance on AllWeatherNight dataset.

Method	Rain Scene		Snow Scene		MACs (G) ↓	Params (M) ↓	Latency (ms) ↓
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑			
NAFNet [39]	22.07	0.728	19.45	0.655	63.2	115.98	612.14
CGNet [40]	21.03	0.697	19.99	0.677	61.7	119.22	722.31
SWSformer-S (Ours)	**23.51**	**0.763**	**23.72**	**0.767**	29.7	49.13	571.69

TABLE 6. Comparison of denoising performance on SIDD dataset with large number of epochs.

Method	PSNR ↑	SSIM ↑	MACs (G) ↓	Params (M) ↓	Latency (ms) ↓
MPRNet [3]	39.71	0.958	573.2	15.74	1854.75
SRMNet [5]	39.72	0.959	284.9	37.59	978.12
Uformer [10]	39.89	0.960	85.3	50.88	794.54
Restormer [12]	40.02	0.960	141.0	26.11	2295.89
NAFNet [13]	40.30	0.962	63.2	115.98	612.14
CGNet [14]	40.39	**0.964**	61.7	119.22	722.31
SWSformer (Ours)	**40.64**	0.963	132.8	194.98	1435.65

information, we investigate the most effective positional encoding method for our model.

- Feedforward Network (FFN): As recent image restoration works [10], [12] have proposed two effective FFNs for Transformers, we determine which is more suitable for SWSformer.
- Channel Attention (CA): Given its adoption in several recent methods [12], [13], [14], we examine the effect of incorporating CA into SWSformer.

For all ablation studies, unless otherwise specified, models are trained on the SIDD dataset for 20 epochs with a batch size of 6, using the same settings as described previously.

1) WINDOW SIZE AND SUBWINDOW SIZE

To determine the optimal window and subwindow sizes for SWS-SA, we evaluate several configurations. In addition, to investigate whether the optimal window size differs with image resolution, we conduct the same experiments at three different resolutions. To generate images of varying resolutions, we crop the center of the original image and then train and infer for each resolution. The results are presented in Table 7, which indicates that a window size of 16×16 paired with a subwindow size of 8×8 achieves the best performance with a resolution of 256×256. Consequently, we adopt this combination as the default configuration for SWS-SA. Furthermore, as shown in Table 7, the optimal window

FIGURE 8. Multi-weather nighttime image restoration results on AllWeatherNight dataset.

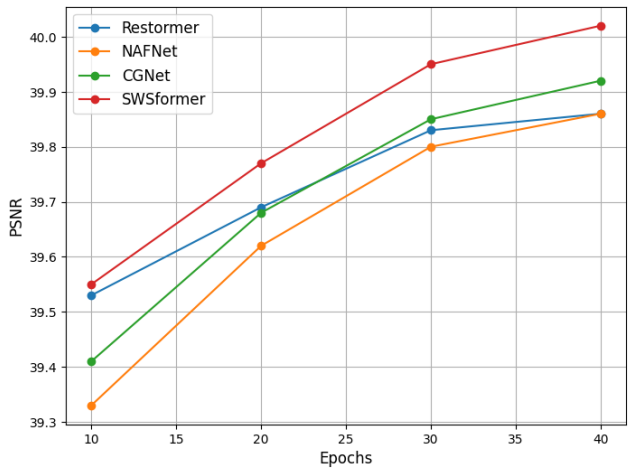


FIGURE 9. Comparison of how model performance scales with training duration on SIDD dataset.

size configuration varies with image resolution. Therefore, to maximize the performance of SWSformer, we recommend that an appropriate window size configuration be set for the target image resolution.

2) AVERAGE POOLING OF SWS-SA

We conduct an ablation study to determine the optimal placement of average pooling in SWS-SA. We compare three variants: applying pooling to the query, applying it to the key/value pair, and using no pooling. To prevent out-of-memory errors in the no-pooling variant, we use a specialized setup for this experiment: a batch size of 2, a window size of 8, and a subwindow size of 8.

The results in Table 8 show a clear outcome. Whereas applying pooling to the key/value pair degrades performance, applying it only to the query maintains the same level of performance while drastically reducing computational cost. We attribute the minimal performance degradation with query average pooling to an inherent property of image data: neighboring pixels tend to contain highly correlated information. This ensures that even after pooling, the attention scores remain largely consistent. As a result, the quality of the aggregated features is maintained, which prevents performance degradation. Conversely, the performance drop observed with key/value average pooling is likely due to the loss of detailed information during the aggregation process. Based on this finding, SWS-SA employs query average pooling.

3) EFFECT OF SWS-SA

To validate the effectiveness of SWS-SA, we compare the performance of SWSformer with and without it. The results in Table 9 clearly demonstrate that SWS-SA significantly improves the model's performance.

4) SWS-SA VS WS-SA

To demonstrate that SWS-SA is an improved version of the Shuffle Transformer's window shuffle self-attention

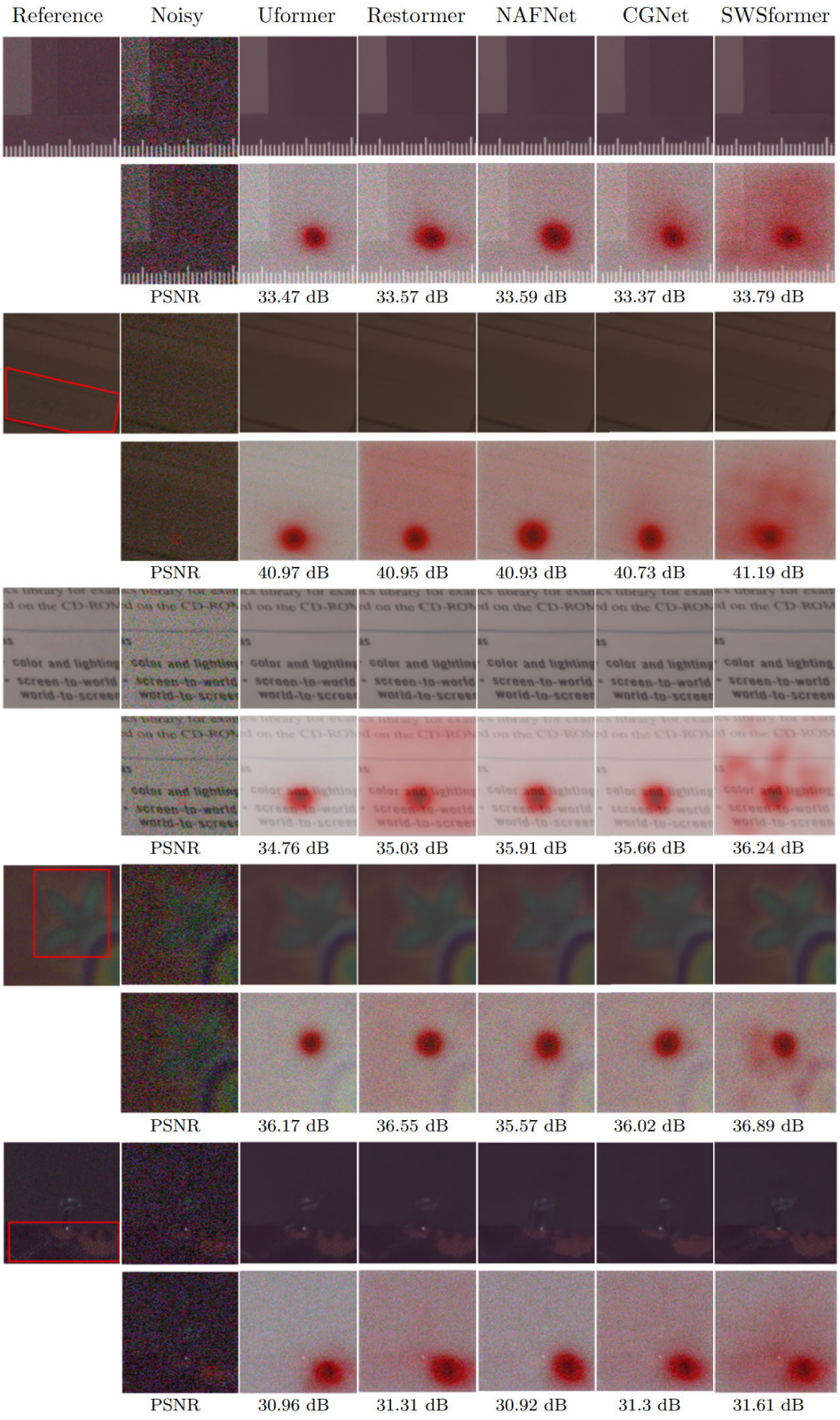


FIGURE 10. Denoising results on SIDD dataset.

(WS-SA) [16], we conduct a comparative experiment. To ensure a fair comparison, all modules apart from the attention mechanism are kept identical, and the window size for both is set to 16.

TABLE 7. Comparison results of various window size and subwindow size patterns.

Image resolution	Window size	Subwindow size	PSNR ↑	MACs (G) ↓
64 × 64	8 × 8	8 × 8	39.37	7.6
		16 × 16	39.38	7.6
	16 × 16	8 × 8	39.38	8.6
		16 × 16	**39.40**	8.6
128 × 128	8 × 8	8 × 8	**39.71**	29.4
		16 × 16	39.67	29.4
	16 × 16	8 × 8	**39.71**	34.1
		16 × 16	39.70	34.1
256 × 256	8 × 8	8 × 8	39.90	111.5
		16 × 16	39.90	111.4
	16 × 16	8 × 8	**39.92**	132.8
		16 × 16	39.89	132.8

TABLE 8. Comparison results between query average pooling, key/value average pooling, and no average pooling in SWS-SA.

Average pooling	PSNR ↑	MACs (G) ↓	Params (M) ↓
×	39.91	393.7	195.0
query	39.91	111.5	195.0
key/value	39.86	111.5	195.0

TABLE 9. Comparison results with and without subwindow shuffle self-attention (SWS-SA).

SWS-SA	PSNR ↑	MACs (G) ↓	Params (M) ↓
×	39.72	84.64	132.2
✓	39.92	132.8	195.0

As shown in Table 10, our SWS-SA achieves better performance at the same computational complexity within the attention mechanism as WS-SA. Note: The total computational complexity of the SWS-SA based model is slightly lower than that of the WS-SA based model. This is because the average pooling in SWS-SA reduces the number of tokens, which in turn lowers the computational load of subsequent layers, such as the projections.

TABLE 10. Comparison results between Shuffle Transformer's window shuffle self-attention (WS-SA) [16] and our subwindow shuffle self-attention (SWS-SA).

Attention	PSNR ↑	MACs (G) ↓	Params (M) ↓
WS-SA	39.86	144.1	195.0
SWS-SA	39.92	132.8	195.0

5) SWSFORMER BLOCK VS SHUFFLE TRANSFORMER BLOCK

To compare the image restoration performance of our SWSformer against the original Shuffle Transformer [16], we conduct a comparative experiment. For this experiment, we create a baseline by replacing the blocks in our SWSformer architecture with the Shuffle Transformer blocks, implemented using the official GitHub code [43]. The configuration for the Shuffle Transformer blocks follows the default settings from the original paper.

The results, presented in Table 11, demonstrate that our SWSformer significantly outperforms the Shuffle Transformer as an image restoration method.

TABLE 11. Comparison of results between the Shuffle Transformer (Shuffle T) block [16] and the SWSformer block.

Block	PSNR ↑	MACs (G) ↓	Params (M) ↓
Shuffle T block	25.18	139.9	195.0
SWSformer block	39.92	132.8	195.0

6) EFFECT OF NWC

We analyze the impact of the Neighbor Window Connection (NWC) module [16] by comparing our model's performance with and without it. The results in Table 12 indicate that including NWC yields a notable performance improvement at the cost of a slight increase in computation. Based on this favorable trade-off, we incorporate NWC into the final SWSformer architecture.

TABLE 12. Comparison results with and without Neighbor Window Connection module (NWC) [16].

NWC	PSNR ↑	MACs (G) ↓	Params (M) ↓
×	39.85	131.6	194.7
✓	39.92	132.8	195.0

7) POSITIONAL ENCODING

To identify the optimal positional encoding method for SWS-SA, we conduct a comparative experiment on three variants: relative position bias (RPB) [11], the Position Encoding Generator (PEG) [17], and no positional encoding. The results, presented in Table 13, show that PEG achieves the highest performance. In contrast, RPB results in lower performance than using no positional encoding. We hypothesize that this performance degradation from RPB is due to query average pooling, which makes it difficult to capture relative positions accurately. Therefore, we adopt PEG as the positional encoding method for SWS-SA.

TABLE 13. Comparison of three positional encoding (PE) methods: relative position bias (RPB) [11], the Position Encoding Generator (PEG) [17], and no positional encoding.

PE	PSNR ↑	MACs (G) ↓	Params (M) ↓
×	39.90	132.4	194.8
RPB	39.85	132.4	242.4
PEG	39.92	132.8	195.0

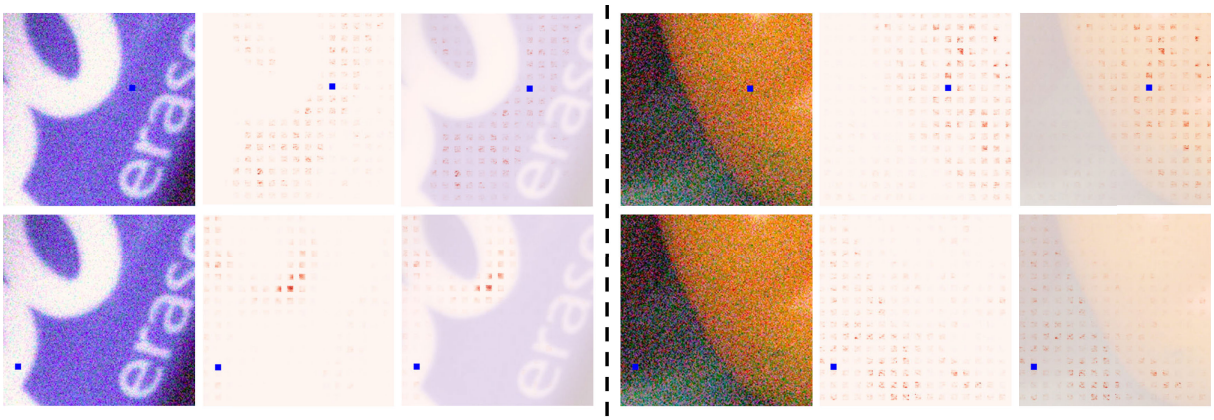


FIGURE 11. Visualization results of attention maps for two types of images. For each image, we visualize the attention scores for queries at two different positions. Left: The target image. Middle: The attention map. Right: The target image overlaid with the attention map. The blue dot represents the query, and the red areas indicate high attention scores.

TABLE 14. Comparison results of two different feedforward networks (FFN). LeFF: Locally enhanced FFN [10], GDFN: Gated Dconv FFN [12].

FFN	PSNR ↑	MACs (G) ↓	Params (M) ↓
LeFF	39.90	132.6	195.3
GDFN	39.92	132.8	195.0

TABLE 15. Comparison results with and without channel attention (CA). In this experiment, we apply NAFNet's Simplified Channel Attention [13] to the output of SWS-SA.

CA	PSNR ↑	MACs (G) ↓	Params (M) ↓
×	39.92	132.8	195.0
✓	39.88	132.8	210.7

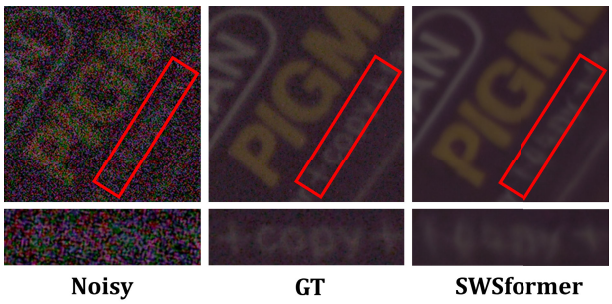


FIGURE 12. Case where SWSformer struggles with restoration.

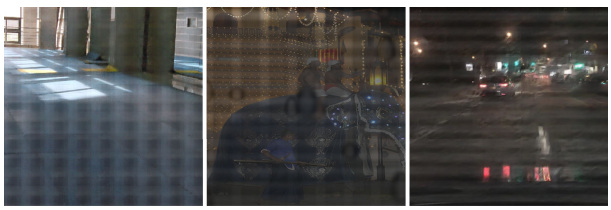


FIGURE 13. The grid artifacts observed in SWSformer's restoration results. Left: Image from the LOLv2 dataset. Middle and Right: Images from the AllWeatherNight dataset (rain scene and snow scene, respectively).

8) FEEDFORWARD NETWORK

We conduct an ablation study to determine the optimal feedforward network (FFN) for SWSformer, comparing two prominent methods: the Gated-Dconv FFN (GDFN) [12] and the Locally-enhanced FFN (LeFF) [10]. The comparison in Table 14 shows that GDFN outperforms LeFF. Consequently, we select GDFN for the SWSformer architecture.

9) CHANNEL ATTENTION

Although channel attention (CA) is used in many recent image restoration methods [12], [13], [14] for its computational efficiency, we investigate its actual effect on SWSformer. To do so, we incorporate NAFNet's Simplified Channel Attention (SCA) [13] after the SWS-SA module and compare performance with a baseline model that omits it.

The results in Table 15 indicate that adding SCA provides no performance benefit. Consequently, we do not include CA in the final SWSformer design.

V. VISUALIZATION OF ATTENTION MAP

To demonstrate that SWSformer captures long-range dependencies, we visualize the attention maps. For this visualization, we use the attention scores extracted from the final SWS-SA of SWSformer. The visualization results of the attention maps for two different images are shown in Fig. 11. From Fig. 11, it's clear that SWSformer captures long-range dependencies in both images. Furthermore, the attended locations change depending on the query's position, indicating that SWSformer can appropriately identify relevant areas.

VI. LIMITATIONS AND FUTURE WORK

Our SWSformer has limitations in restoration for cases that require pixel-level long-range dependencies. For example, in Fig. 12, the small characters are heavily corrupted by noise, and to restore them effectively, the model needs to capture the entire character at a pixel level. However, SWS-SA applies average pooling to the query for efficiency, which prevents SWSformer from capturing pixel-level long-range dependencies. As a result, SWSformer fails to properly restore the characters in Fig. 12. To achieve effective restoration in such cases, our future work will focus on proposing an attention mechanism that can efficiently capture

pixel-level long-range dependencies. Furthermore, the grid artifacts are observed in several restoration results for both the LOLv2 dataset and AllWeatherNight dataset, as shown in Fig. 13. It has been found that these grid artifacts are not entirely resolvable simply by expanding the kernel size of the NWC or by employing overlapping window partitioning. This issue is likely attributable to SWS-SA's difficulty in identifying high-relevance information under conditions of low illumination or dense haze. Therefore, an effective strategy for improvement is the incorporation of prior guidance [54] to the SWS-SA. Regarding efficiency, although our SWSformer is more efficient than other Transformer models, its application to mobile devices and real-time processing is difficult, especially due to the large number of parameters. Therefore, we aim to create a lightweight version of SWSformer in future research by using techniques such as pruning and quantization.

VII. CONCLUSION

In this work, we proposed subwindow shuffle self-attention (SWS-SA), an efficient mechanism for capturing long-range dependencies. SWS-SA achieves a large receptive field by using a subwindow-based spatial shuffle, while keeping computational complexity low by applying average pooling to the query. Building on this, we developed the Subwindow Shuffle Transformer (SWSformer), a new architecture that integrates SWS-SA with effective techniques from related studies. We evaluated SWSformer on image denoising, deblurring, and deraining tasks, where it demonstrated state-of-the-art performance.

AVAILABILITY OF DATA

All datasets used for supporting the conclusions of this article are available from the public data repository at the website listed below.

- SIDD dataset: https://abdokamel.github.io/sidd/
- GoPro dataset: https://seungjunnah.github.io/Datasets/gopro.html
- Rain100H dataset: https://www.kaggle.com/datasets/nguynhoitrung/rain100h
- LOLv2 dataset: https://www.kaggle.com/datasets/tanhyml/lol-v2-dataset
- AllWeatherNight dataset: https://huggingface.co/datasets/YuetongLiu/AllWeatherNight

REFERENCES

[1] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Learning enriched features for real image restoration and enhancement," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 492–511.

[2] S. Cheng, Y. Wang, H. Huang, D. Liu, H. Fan, and S. Liu, "NBNet: Noise basis learning for image denoising with subspace projection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4894–4904.

[3] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14816–14826.

[4] L. Chen, X. Lu, J. Zhang, X. Chu, and C. Chen, "HINet: Half instance normalization network for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 182–192.

[5] C.-M. Fan, T.-J. Liu, K.-H. Liu, and C.-H. Chiu, "Selective residual M-Net for real image denoising," in *Proc. 30th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2022, pp. 469–473.

[6] K. Zhang, W. Luo, Y. Zhong, L. Ma, W. Liu, and H. Li, "Adversarial spatio-temporal learning for video deblurring," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 291–301, Jan. 2019.

[7] W. Yu, Z. Huang, W. Zhang, L. Feng, and N. Xiao, "Gradual network for single image de-raining," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1795–1804.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 5998–6008.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–16.

[10] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general U-shaped transformer for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17662–17672.

[11] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[12] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5718–5729.

[13] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 17–33.

[14] A. Ghasemabadi, M. K. Janjua, M. Salameh, C. Zhou, F. Sun, and D. Niu, "Cascadedgaze: Efficiency in global context extraction for image restoration," *Trans. Mach. Learn. Res. (TMLR)*, pp. 1–20, 2024.

[15] J. Gu and C. Dong, "Interpreting super-resolution networks with local attribution maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9195–9204.

[16] Z. Huang, Y. Ben, G. Luo, P. Cheng, G. Yu, and B. Fu, "Shuffle transformer: Rethinking spatial shuffle for vision transformer," 2021, *arXiv:2106.03650*.

[17] X. Chu, Z. Tian, B. Zhang, X. Wang, X. Wei, H. Xia, and C. Shen, "Conditional positional encodings for vision transformers," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–19.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[19] K. Zhang, W. Ren, W. Luo, W.-S. Lai, B. Stenger, M.-H. Yang, and H. Li, "Deep image deblurring: A survey," *Int. J. Comput. Vis.*, vol. 130, no. 9, pp. 2103–2130, Sep. 2022.

[20] X. Chen, J. Pan, J. Dong, and J. Tang, "Towards unified deep image deraining: A survey and a new benchmark," 2023, *arXiv:2310.03535*.

[21] N. Gao, X. Jiang, X. Zhang, and Y. Deng, "Efficient frequency-domain image deraining with contrastive regularization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024, pp. 240–257.

[22] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, San Francisco, CA, USA, Tech. Rep., 2018.

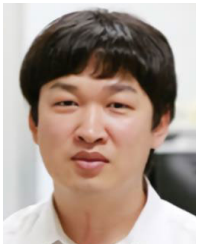
[23] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2018, pp. 4171–4186.

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 10347–10357.

[25] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. H. Tay, J. Feng, and S. Yan, "Tokens-to-Token ViT: Training vision transformers from scratch on ImageNet," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 538–547.

[26] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 548–558.
[27] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Álvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 12077–12090.
[28] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6877–6886.
[29] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213–229.
[30] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–16.
[31] F. Long, Z. Qiu, Y. Pan, T. Yao, J. Luo, and T. Mei, "Stand-alone inter-frame attention in video models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 68–80.
[32] Z. Zhang, H. Zhang, L. Zhao, T. Chen, S. Ö. Arik, and T. Pfister, "Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, 2022, pp. 3417–3425.
[33] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal attention for long-range interactions in vision transformers," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 30008–30022.
[34] X. Pan, T. Ye, Z. Xia, S. Song, and G. Huang, "Slide-transformer: Hierarchical vision transformer with local self-attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 2082–2091.
[35] D. Hendrycks and G. Gimpel, "Gaussian error linear units (GELUs)," 2023, *arXiv:1606.08415*.
[36] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "SwinIR: Image restoration using Swin transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1833–1844.
[37] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[38] S. W. Zamir. (2022). *Restormer*. [Online]. Available: https://github.com/swz30/Restormer
[39] L. Chen. (2022). *NAFNet*. [Online]. Available: https://github.com/megvii-research/NAFNet
[40] A. Ghasemabadi. (2024). *CGNet & CGNet-MH-Local*. [Online]. Available: https://github.com/Ascend-Research/CascadedGaze
[41] X. Chu. (2022). *Restormer-Local*. [Online]. Available: https://github.com/megvii-research/TLC
[42] S. Liu. (2022). *NAFNet-MH-Local*. [Online]. Available: https://github.com/Liu-SD/multi-output-deblur
[43] W. Zhang. (2021). *Shuffle-Transformer*. [Online]. Available: https://github.com/mulinmeng/Shuffle-Transformer
[44] N. Gao. (2024). *FADformer*. [Online]. Available: https://github.com/deng-ai-lab/FADformer
[45] M. V. Conde. (2025). *DarkIR*. [Online]. Available: https://github.com/cidautai/DarkIR
[46] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1692–1700.
[47] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 257–265.
[48] X. Chu, L. Chen, C. Chen, and X. Lu, "Improving image restoration by revisiting global information aggregation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 53–71.
[49] S. Liu, P. Qiao, and Y. Dou, "Multi-outputs is all you need for deblur," 2022, *arXiv:2208.13029*.
[50] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1685–1694.
[51] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3932–3941.
[52] W. Yang, W. Wang, H. Huang, S. Wang, and J. Liu, "Sparse gradient regularized deep retinex network for robust low-light image enhancement," *IEEE Trans. Image Process.*, vol. 30, pp. 2072–2086, 2021.
[53] D. Feijoo, J. C. Benito, A. Garcia, and M. V. Conde, "DarkIR: Robust low-light image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2025, pp. 10879–10889.
[54] Y. Liu, Y. Xu, Y. Wei, X. Bi, and B. Xiao, "Clear nights ahead: Towards multi-weather nighttime image restoration," 2025, *arXiv:2505.16479*.

NAGAYUKI OKITSU received the B.E. degree from Shimane University, Japan, in 2024, where he is currently pursuing the M.E. degree. His undergraduate research was on image restoration using deep learning. His current research interests include deep learning, with applications in image processing, time-series processing, and natural language processing.



MASATO SHIRAI received the B.E., M.E., and Ph.D. degrees from Hosei University, Japan, in 2011, 2013, and 2016, respectively. He was a Collaborative Researcher with the Micro/Nano Technology Research Center, Hosei University, from 2016 to 2017. He then joined a Faculty Member with Shimane University, where he was a Project Assistant Professor, from 2017 to 2019, and an Assistant Professor, from 2019 to 2024. He is currently an Associate Professor with Shimane University. His research interests include image processing and natural language processing.