Random Shuffle Transformer for Image Restoration

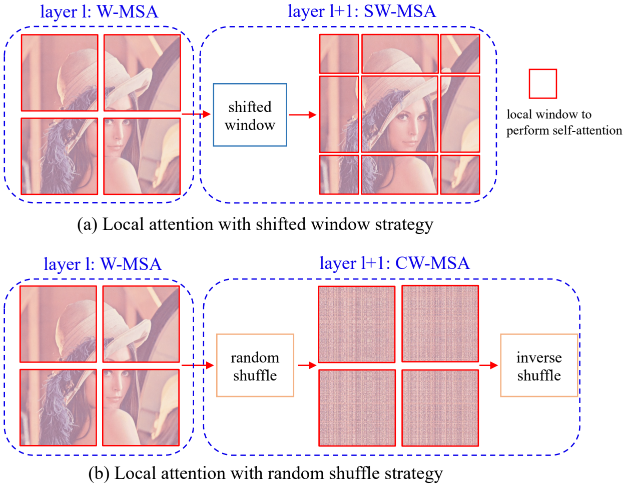
Jie Xiao [1] Xueyang Fu [1] Man Zhou [1] Hongjian Liu [1] Zheng-Jun Zha [1]

Abstract

Non-local interactions play a vital role in boosting performance for image restoration. However, local window Transformer has been preferred due to its efficiency for processing high-resolution images. The superiority in efficiency comes at the cost of sacrificing the ability to model non-local interactions. In this paper, we present that local window Transformer can also function as modeling non-local interactions. The counter-intuitive function is based on the permutation-equivariance of self-attention. The basic principle is quite simple: by *randomly shuffling* the input, local self-attention also has the potential to model non-local interactions without introducing extra parameters. Our random shuffle strategy enjoys elegant theoretical guarantees in extending the local scope. The resulting Transformer dubbed *ShuffleFormer* is capable of processing high-resolution images efficiently while modeling non-local interactions. Extensive experiments demonstrate the effectiveness of ShuffleFormer across a variety of image restoration tasks, including image denoising, deraining, and deblurring. Code is available at https://github.com/jiexiaou/ShuffleFormer.

1. Introduction

Image restoration is typically an ill-posed inverse problem aiming to reconstruct the high-quality clean image from its low-quality counterpart, which lays the foundation for various vision tasks. According to the degradation process, image restoration can be categorized into many subtasks, *e.g.*, denoising, super-resolution, deblurring, deraining and compression artifact reduction. With the popularity of deep learning (LeCun et al., 2015), deep neural networks, especially convolutional neural networks (CNNs), have pushed the state-of-the-art for various image restoration tasks. Recently, vision transformers (Dosovitskiy et al., 2021; Liu et al., 2021) have achieved comparable or even better performance across a number of vision tasks. Similarly, there are also numerous literature studying how to make Transformer suitable for these pixel-level regression tasks. IPT (Chen et al., 2021) adopts the standard Transformer architecture (Vaswani et al., 2017), which conducts global self-attention across spatial dimension, for image processing. IPT enjoys the typical advantage—modeling global interactions—of the standard Transformer. However, IPT inevitably suffers from quadratic complexity in both computation and memory usage, which hinders its application to high-resolution images. Local window Transformers, *e.g.*, SwinIR (Liang et al., 2021) and Uformer (Wang et al., 2022), first partition the input into non-overlapping local windows, and restrict computation of attention in a local window. In order to enable inter-window interactions, shifted window strategy (Liu et al., 2021; Tolstikhin et al., 2021) is employed, which, however, still cannot escape the local scope brought by window partition. The complexity of local window Transformer is linear to the input resolution, making it more suitable for processing high-resolution images. In conclusion, there exists an unpleasing trade-off between modeling non-local interactions and efficiency in



(a) Local attention with shifted window strategy

(b) Local attention with random shuffle strategy

Figure 1: The comparison between shifted window strategy and our proposed random shuffle strategy.

[1] University of Science and Technology of China, Hefei, China. Correspondence to: Xueyang Fu <xyfu@ustc.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

computation and memory usage. Therefore, a natural question arises: is it possible to model *non-local* interactions with *local window* Transformer? In this work, we provide a solution to the seemingly counterintuitive question.

The reason behind local self-attention decreases the cost of computation and memory is that window partition limits the number of tokens in self-attention. Consequently, linear complexity can be retained as long as window partition before self-attention is reserved. The challenge we now face is to let self-attention capture non-local interactions with only local window configuration. As aforementioned, shifted window strategy cannot extend the local scope. Therefore, it is possible to accomplish our goal—using local self-attention to capture non-local interactions—with only shifted window strategy replaced. Since the configuration of window partition is reserved, in order to model non-local interactions, we have to relieve the constraint of locality brought by window partition. It is ready to approach an elegant solution: we choose to randomly shuffle the input across spatial dimension instead of adopting the shifted window strategy. The local window based multi-head self-attention layer is termed window multi-head self-attention (W-MSA); W-MSA with random shuffle is named chaotic window multi-head self-attention (CW-MSA) [1], and with shifted window is called shifted window multi-head self-attention (SW-MSA). Since shuffling the pixels in space will destroy semantic information, after passing through CW-MSA, a corresponding inverse shuffle must be conducted to exactly recover the origin rearrangement of pixels. Random shuffle and inverse shuffle constitute an invertible transformation pair without introducing *any* information loss. The comparison between our random shuffle strategy and shifted window strategy is illustrated in Figure 1.

A potential concern is that random shuffle may cause instability for training process. Indeed, the permutation-equivariance property of self-attention, *i.e.*, it gives the same output independently of how the input tokens are shuffled (Cordonnier et al., 2019), ensures that Transformer exhibits robustness to random permutation. For training, each individual input goes through CW-MSA with an independent random shuffle sample, which introduces no extra parameters or FLOPs. In this way, non-local interactions can be captured from the sense of expectation. During testing, in the similar spirit with Dropout (Srivastava et al., 2014), we approximate the expectation of CW-MSA by Monte-Carlo averaging. This ensures that for each CW-MSA, the actual output at testing time approximates the expected output.

In conclusion, the contributions of this work include:

- We propose the chaotic window multi-head self-attention (CW-MSA) as an alternative to shifted window multi-head self-attention (SW-MSA) for image restoration. CW-MSA can extend the local scope of SW-MSA without introducing extra parameters. CW-MSA is the first to successfully model non-local interactions only using local self-attention. Our method also enjoys elegant theoretical guarantees;
- We design the training and testing strategy for CW-MSA. For training, each input is passed through CW-MSA with an independent sample of random shuffle. During testing, we approximate the layer-wise expectation by Monte-Carlo averaging;
- Extensive experiments on various tasks, *e.g.*, deraining, denoising and deblurrring, validate that the restoration performance can be consistently improved by equipping with our chaotic window self-attention.

2. Related works

2.1. Image Restoration

Image Restoration (Zhang et al., 2019c; 2017b; Fan et al., 2020; Luo et al., 2021; Liu et al., 2018; Li et al., 2021b;a; Lin et al., 2022b) is typically an ill-posed inverse problem, which aims to reconstruct the latent clean image from the degraded counterpart. Classic image restoration tasks include image denoising (Laine et al., 2019; Yue et al., 2019; Zhang et al., 2018b; 2017a; Dai et al., 2021; Zhang et al., 2021a), image super-resolution (Dai et al., 2019; Haris et al., 2018; Huang et al., 2020; Zhang et al., 2021b), image deblurring (Dong et al., 2020; Nah et al., 2017; Kupyn et al., 2018; Zhang et al., 2018a; Ren et al., 2021; 2018), image deraining (Fu et al., 2017; Zhang et al., 2019b; Ren et al., 2019), etc. Recently, remarkable progress against traditional model-driven methods (Dabov et al., 2007a; Wang et al., 2008; He et al., 2011) has been achieved due to the development of deep learning technologies (LeCun et al., 2015). Instead of relying on handcraft priors, learning based methods directly learn to project from noisy to clean ones from pre-collected data in supervised or self-supervised way (Lehtinen et al., 2018; Batson & Royer, 2019). Non-local prior (Dabov et al., 2007b) has been proven effective for image restoration. Modeling non-local interactions is a key ingredient for the advanced models to achieve promising results. Researchers have made massive efforts elaborating sophisticated models to capture non-local interactions, including non-local methods (Lefkimmiatis, 2017; Plötz & Roth, 2018), graph methods (Mou et al., 2021), and Transformers (Chen et al., 2021).

2.2. Vision Transformer

With great success of Transformer (Vaswani et al., 2017; Devlin et al., 2018) in NLP, Vision Transformers (Dosovit-

[1] Chaotic window multi-head self-attention is named because the participant pixels are randomly rearranged.

skiy et al., 2021; Liu et al., 2021; Wu et al., 2022; Nguyen et al., 2022; Lin et al., 2022a) have also gained popularity in vision community. ViT (Dosovitskiy et al., 2021) treated image patches as token sequence and applied the vanilla transformer on it for image classification. Swin Transformer (Liu et al., 2021) brought in the locality prior to self-attention and adopted the shifted window self-attention to establish a hierarchical architecture. A few transformers (Chen et al., 2021; Liang et al., 2021; Wang et al., 2022; Zamir et al., 2022) for low-level vision have also arose. Nevertheless, most of them adopt the vanilla global self-attention or shifted window self-attention, which cannot model non-local interactions with linear complexity.

3. Method

We first recall the mathematical formulation of self-attention layers, comprising the vanilla global self-attention and the local self-attention variant. The global self-attention suffers from quadratic complexity in computation and memory usage while the local self-attention cannot model non-local interactions. Then, we introduce the chaotic window self-attention to enjoys both merits of efficiency and non-local interactions modeling. Last, we construct the ultimate Transformer model, named ShuffleFormer, by alternating local window self-attention layer and chaotic window self-attention layer for image restoration.

3.1. Self-Attention

Global Self-Attention. Self-attention can be described as mapping a query and a set of key-value pairs to an output, where the query, keys and values are obtained from linear projections of input. One of the widely-used implementations is the scaled dot-product attention, as expressed in Equation (1). Let x denote the input. SA first computes the query Q, key K and value V through linear projections parameterized by W^Q, W^K, and W^V respectively. Then, a similarity matrix is computed by the scaled dot-production of the query and key, where the scaled factor is the root of dimension of key. The similarity matrix is further normalized by softmax function to produce the weight, which is used to aggregate the value. A key property of SA is that it cannot make use of positional information. To compensate for that, a number of position encoding approaches are studied, *e.g.*, absolute positional encoding (Vaswani et al., 2017), relative positional encoding (Shaw et al., 2018; Bello et al., 2019; Liu et al., 2021), etc. To enhance representation, SA is often extended to the multi-head version (Vaswani et al., 2017).

$$\text{SA}\left(x\right) = \text{softmax}\left(\frac{xW^Q(xW^K)^T}{\sqrt{d_k}}\right)xW^V. \tag{1}$$

Self-attention is often used to model global interactions. However, the complexity in time and memory is quadratic with respect to the token number. For image restoration, the token number, which often corresponds to the input pixels, incurs prohibitively huge burden in both computation and memory cost. In addition, due to the lack of inductive biases, the vanilla Transformer (Dosovitskiy et al., 2021; Chen et al., 2021) requires expensive pretraining on large-scale dataset.

Local Self-Attention. Locality (LeCun et al., 1989), as a widely acknowledged prior for vision tasks, is incorporated in self-attention. Local attention (Ramachandran et al., 2019; Liu et al., 2021) restricts the scope of participant tokens to a local window. Swin Transformer (Liu et al., 2021) adopts window self-attention (W-SA) to exploit the locality prior and reduce the complexity as well. To enable communications between windows, Swin Transformer adopted the shifted window strategy, which produces the shifted window self-attention (SW-SA). By restricting self-attention to the local scope, the complexity of Swin Transformer is reduced to linearity w.r.t. the input resolution. The mathematical formulation of W-SA/SW-SA is

$$\begin{aligned}\text{W-SA}\left(x\right) &= \text{SA}\left(\text{W}(x)\right),\\ \text{SW-SA}\left(x\right) &= \text{SA}\left(\text{SW}(x)\right),\end{aligned} \tag{2}$$

where the $\text{W}(x)$/$\text{SW}(x)$ function partitions x into non-overlapping windows. Despite the significant reduction in complexity, local attention sacrifices the capacity of modeling non-local relationships. Non-local relationships are essential for image restoration since, in general, degradation often appears the non-local characteristic. Therefore, given the consideration of both the complexity and scope size, it is desirable to utilize local attention to accomplish the goal of modeling non-local interactions.

3.2. Chaotic Window Self-Attention

3.2.1. Shuffling breaks the local barrier

Our objective is to design an alternative strategy to shifted window strategy so that the local attention is capable of modeling non-local interactions. In this way, the strategy can function as a plug-and-play way to extend the scope of local attention. Our solution is somewhat ambitious: *randomly shuffling* the input across spatial dimension before passing though local attention. Although the subsequent attention is still restricted within a fixed window, the participant tokens come from the non-local field, which breaks the local barrier brought by local attention. Random shuffle thoroughly destroys the relative relationships between pixels.

To achieve the purpose of modeling non-local interactions with *only* local window based self-attention, we first assume the size of local window is *much smaller* compared with the input resolution (Assumption 3.1).

Assumption 3.1 (Local Partition)**.** For local window partition for performing self-attention, the size of local window

$s \times s$ satisfies $s \ll H$ and $s \ll W$, where $H \times W$ is the input resolution.

Definition 3.2 (Random Shuffle Function)**.** $\mathcal{S} : X \mapsto Y$ is the random shuffle function if the elements of $Y \in \mathbb{R}^{H \times W}$ correspond to a random permutation of the elements of $X \in \mathbb{R}^{H \times W}$. Let $\mathbf{m}$ be the 2D index of X and $\mathtt{m}$ be the corresponding index of Y, that is, $Y_{\mathtt{m}} = \mathcal{S}(X)_{\mathtt{m}} = X_{\mathbf{m}}$.

Although window self-attention only captures relationships of pixels within the same local window, random shuffle may pull two pixels into the same window regardless the distance. To depict the property formally, we have Definition 3.3:

Definition 3.3 (Chaotic Set)**.** We denote the collection of pixel pairs that locate within the same local window after performing the random shuffle as the chaotic set $\mathcal{C}_{\mathcal{S}}$, *i.e.*, $\mathcal{C}_{\mathcal{S}} = \{(\mathbf{m}_1, \mathbf{m}_2)$: *after performing the random shuffle* $\mathcal{S}$*, pixels of index* $\mathtt{m}_1$ *and* $\mathtt{m}_2$ *are in the same local window*$\}$.

Random shuffle allows self-attention to integrate information across the whole image. To depict the capacity of modeling non-local interactions, we define the distance criterion between a pixel pair in the following:

Definition 3.4 (Chaotic Distance)**.** We define the chaotic distance between pixels of index $\mathbf{m}_1$ and $\mathbf{m}_2$ as

$$d_{\mathcal{S}}(\mathbf{m}_1, \mathbf{m}_2) = \begin{cases} \|\mathbf{m}_1 - \mathbf{m}_2\|_2, & \text{if } (\mathbf{m}_1, \mathbf{m}_2) \in \mathcal{C}_{\mathcal{S}}, \\ \infty, & \text{if } (\mathbf{m}_1, \mathbf{m}_2) \notin \mathcal{C}_{\mathcal{S}}. \end{cases} \tag{3}$$

In Equation (3), infinity means that the interaction cannot be captured and *does not* affect derivation of modeling distance. We denote the expected chaotic distance to pixel $\mathbf{m}_1$ as $d(\mathbf{m}_1)$, which is defined by:

$$d(\mathbf{m}_1) = \mathop{\mathbb{E}}_{\mathbf{m}_2 | (\mathbf{m}_1, \mathbf{m}_2) \in \mathcal{C}_{\mathcal{S}}} \left[d_{\mathcal{S}}(\mathbf{m}_1, \mathbf{m}_2) \right]. \tag{4}$$

$d(\mathbf{m}_1)$ in Definition 3.4 measures the expected distance between a pixel pair captured by CW-SA. We can prove that the asymptotic lower bound of $d(\mathbf{m}_1)$ is $\Omega(\frac{\sqrt{2}}{4}(H + W))$. Rigorously, we have the following:

Theorem 3.5. *The lower bound of* $d(\mathbf{m}_1)$ *is*

$$\begin{aligned} &\frac{\sqrt{2}}{4(HW-1)} [H m_{w_1}(m_{w_1}+1) + W m_{h_1}(m_{h_1}+1) + \\ &H(W - m_{w_1})(W - m_{w_1} - 1) + W(H - m_{h_1})(H - m_{h_1} - 1)], \end{aligned} \tag{5}$$

in which $\mathbf{m}_1 = (m_{h_1}, m_{w_1})$.

The proof of Theorem 3.5 is delayed to Appendix A. A corollary of Theorem 3.5 is the following.

Corollary 3.6. *Let* $\bar{d}(\mathbf{m}_1)$ *be the average distance of interactions captured by a window of CW-SA. Then, we have*

$$\bar{d}(\mathbf{m}_1) \approx d(\mathbf{m}_1). \tag{6}$$

Proof. We compute the average distance in a window by:

$$\bar{d}(\mathbf{m}_1) = \frac{1}{s^2 - 1} \sum_{i=1}^{s^2-1} d(\mathbf{m}_1 | HW - i). \tag{7}$$

$d(\mathbf{m}_1 | HW - i)$ denotes the expected chaotic distance when the total number of remaining pixels is $HW - i$. According to Assumption 3.1, when $i \in [2, s^2 - 1]$, we have the approximation:

$$d(\mathbf{m}_1 | HW - i) \approx d(\mathbf{m}_1 | HW - 1) = d(\mathbf{m}_1). \tag{8}$$

Therefore, we can approximate $\bar{d}(\mathbf{m}_1)$ by:

$$\begin{aligned} \bar{d}(\mathbf{m}_1) &= \frac{1}{s^2 - 1} \sum_{i=1}^{s^2-1} d(\mathbf{m}_1 | HW - i) \\ &\approx \frac{1}{s^2 - 1} \sum_{i=1}^{s^2-1} d(\mathbf{m}_1 | HW - 1) \\ &= \frac{1}{s^2 - 1} \sum_{i=1}^{s^2-1} d(\mathbf{m}_1) \\ &= d(\mathbf{m}_1) \end{aligned} \tag{9}$$

which completes the proof. □

According to Corollary 3.6, the average distance of interaction captured by a window of CW-SA is $\Omega(\frac{\sqrt{2}}{4}(H + W))$, thereby capturing non-local interactions. More evidence is included in Appendix E.1. As aforementioned, self-attention cannot model positional relationships. This property seems not preferable for vision tasks since the rearrangement of pixels contains basic structural information for an image. However, this permutation-equivariance of self-attention, in turn, supports the increasing of local scope by randomly shuffling. Since the relative position of pixels is essential for constructing semantic information, the output must be *inversely shuffled* to exactly align with the input. The random shuffle and inverse shuffle constitutes a invertible transformation pair without any information loss. Besides, to compensate for the inability of positional modeling, a simple depth-wise convolution layer is conducted before shuffling (Chu et al., 2023), as shown in Figure 2. The derived attention mechanism, named Chaotic Window Self-Attention (CW-SA), can process high-resolution input in linear complexity as well as model non-local relationships. To enrich the diversity of representation, CW-SA can easily be extended to the multi-head version (CW-MSA). Without loss of generality, we only consider the single-head version for simplicity.

3.2.2. Training with Random Shuffle

Random shuffle is introduced in local attention to break the local barrier. The window partition plays the role of

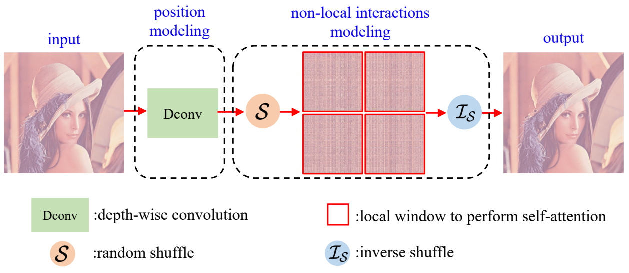

input
position modeling
non-local interactions modeling
output
Dconv
$\mathcal{S}$
$\mathcal{I}_\mathcal{S}$
Dconv :depth-wise convolution
:local window to perform self-attention
$\mathcal{S}$:random shuffle
$\mathcal{I}_\mathcal{S}$:inverse shuffle

Figure 2: Training of CW-SA with a random shuffle sample.

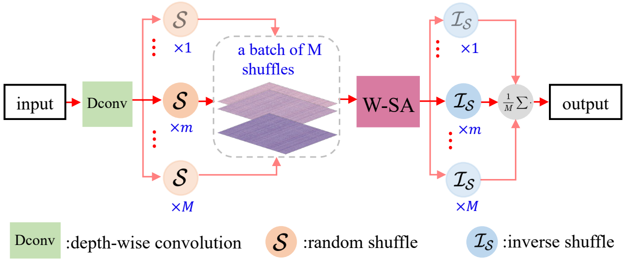

input
Dconv
$\mathcal{S}$
×1
×m
×M
a batch of M shuffles
W-SA
$\mathcal{I}_\mathcal{S}$
×1
×m
×M
$\frac{1}{M}\Sigma$
output
Dconv :depth-wise convolution
$\mathcal{S}$:random shuffle
$\mathcal{I}_\mathcal{S}$:inverse shuffle

Figure 3: Testing of CW-SA with Monte-Carlo averaging.

scaling down the complexity of self-attention. By shuffling randomly, any pair of tokens has the same probability of entering the same local window regardless of the distance, thus providing the foundation for modeling non-local interactions. For notation uncluttered, we use typewriter typestyle $\mathtt{x}$ to stand for the randomly shuffled version of regular feature x. As illustrated in Figure 2, each individual input goes through CW-SA with a random shuffle sample, window self-attention, and the corresponding inverse shuffle. The above procedure can be formulated as:

$$\begin{aligned} \mathtt{x} &= \mathcal{S}\left(x\right), \\ \mathtt{z} &= \text{W-SA}\left(\mathtt{x}\right), \\ \text{CW-SA}\left(x;\mathcal{S}\right) &= \mathcal{I}_\mathcal{S}\left(\mathtt{z}\right), \end{aligned} \tag{10}$$

where $\mathcal{S}(\cdot)$ is the random shuffle function and $\mathcal{I}_\mathcal{S}(\cdot)$ is the corresponding inverse shuffle function. The random shuffle samples are independent along the depth of network. With random shuffle, CW-SA has the same probability to model the interactions between any pair of pixels. Therefore, CW-SA can model non-local interactions from the sense of expectation. It is noteworthy that although possessing the non-local scope, CW-SA can also be trained efficiently due to sampling from random shuffle.

3.2.3. Testing with Monte-Carlo Averaging

We introduce random factors, *i.e.*, random factors of stacked CW-SAs, into Transformer. From the Bayesian perspective, these random factors should be marginalized to yield the ultimate restored result (see Appendix D.1). However, the way of random shuffle forms exponentially many potential models. It is not feasible to explicitly average the predictions from these models. Inspired by Dropout (Srivastava et al., 2014), the expectation of the whole model can be well approximated by the layer-wise expectation. Therefore, it makes sense to compute the expectation with respect to the random shuffle for CW-SA, which is expressed as

$$\text{CW-SA}^{\text{test}}\left(x\right) = \mathop{\mathbb{E}}_{\mathcal{S}}\left[\text{CW-SA}\left(x,\mathcal{S}\right)\right]. \tag{11}$$

However, to compute CW-SA$^{\text{test}}$ according to Equation (11), it is required to enumerate all possible shuffle outcomes, which in practice is prohibitively expensive. An approximation to expectation is Monte-Carlo averaging, as formulated by Equation (12). Specifically, the procedure is: shuffling the input x independently by M times, computing M outputs of the CW-SA and averaging the outputs to obtain the Monte-Carlo estimator. As $M \to \infty$, the Monte-Carlo estimator gets close to the true expectation.

$$\text{CW-SA}^{\text{test}}\left(x\right) \approx \frac{1}{M}\sum_{i=1}^{M}\text{CW-SA}\left(x,\mathcal{S}_i\right). \tag{12}$$

It seems that the testing time will be scaled by M, which is the number of averaged forward passes. However, the multiple forward passes can be conducted concurrently with modern accelerators, which significantly reduces the testing time. Specifically, this acceleration can be done by transferring an input to GPU(s) and setting a mini-batch comprising the same input multiple times. CW-SA shuffles *independently* along the batch dimension. After one forward pass through CW-SA, averaging over the mini-batch yields the Monte-Carlo estimation. Figure 3 illustrates the procedure.

3.2.4. Complexity Analysis

We provide a detailed analysis about the time and space complexity of CW-MSA in Tables 1 and 2. We include the complexity of MSA and W-MSA for comparison. For training, our CW-MSA is as efficient as W-MSA in speed and memory usage, as shown in Table 1. However, W-MSA is unable to capture non-local interactions. MSA can model interactions from the non-local scope while suffering from quadratic complexity in input resolution. In contrast, our CW-MSA enjoys efficient training while possessing the capacity of modeling the non-local interactions. For testing, as shown in Table 2, W-MSA can inference with linear complexity but is still restricted by local interactions. On the other hand, the complexity of CW-MSA for testing is linear to the input resolution and scaled by the number of Monte-Carlo samples compared with the training case.

3.3. ShuffleFormer

The proposed chaotic window self-attention is capable of modeling non-local interactions in linear complexity. For

Table 1: Complexity comparison for training. B: batch size, (H, W, C): feature size, s: local window size, h: number of heads.

Method	Time Complexity	Space Complexity	Non-local Scope?
MSA	$\mathcal{O}(BH^2W^2C + BHWC^2)$	$\mathcal{O}(BhH^2W^2 + BHWC)$	$\surd$
W-MSA	$\mathcal{O}(BHWCs^2 + BHWC^2)$	$\mathcal{O}(BhHWs^2 + BHWC)$	$\times$
CW-MSA	$\mathcal{O}(BHWCs^2 + BHWC^2)$	$\mathcal{O}(BhHWs^2 + BHWC)$	$\surd$

Table 2: Complexity comparison for testing. B: batch size, (H, W, C): feature size, s: local window size, h: number of heads, M: number of Monte-Carlo samples.

Method	Time Complexity	Space Complexity	Non-local Scope?
MSA	$\mathcal{O}(BH^2W^2C + BHWC^2)$	$\mathcal{O}(BhH^2W^2 + BHWC)$	$\surd$
W-MSA	$\mathcal{O}(BHWCs^2 + BHWC^2)$	$\mathcal{O}(BhHWs^2 + BHWC)$	$\times$
CW-MSA	$\mathcal{O}(MBHWCs^2 + MBHWC^2)$	$\mathcal{O}(MBhHWs^2 + MBHWC)$	$\surd$

image restoration tasks, both *local* and *non-local* interactions are essential for recovering clean images. Therefore, the desired model is expected to not only model non-local interactions but also enhance locality. To this end, we alternate W-MSA (for local interactions) and CW-MSA (for non-local interactions) in consecutive blocks, which is computed according to

$$\begin{aligned}\hat{x}^l &= \text{W-MSA}(\text{LN}(x^{l-1})) + x^{l-1},\\ x^l &= \text{MLP}(\text{LN}(\hat{x}^l)) + \hat{x}^l,\\ \hat{x}^{l+1} &= \text{CW-MSA}(\text{LN}(x^l)) + x^l,\\ x^{l+1} &= \text{MLP}(\text{LN}(\hat{x}^{l+1})) + \hat{x}^{l+1},\end{aligned} \tag{13}$$

where $\hat{x}^l$ and x^l denote the output features of the (C)W-MSA module and the MLP module for block l, respectively. Similar to previous works (Liu et al., 2021; Wang et al., 2022), we establish an hierarchical Transformer with the widely-used U-shape architecture (Ronneberger et al., 2015; Isola et al., 2017). The resulting Transformer, named ShuffleFormer, not only enjoys the capacity of modeling non-local interactions but also enhances locality. It should be emphasized ShuffleFormer can process input in linear complexity with respect to the resolution. The overall architecture of ShuffleFormer is illustrated in Figure 4.

4. Experiment

4.1. Experimental Setting

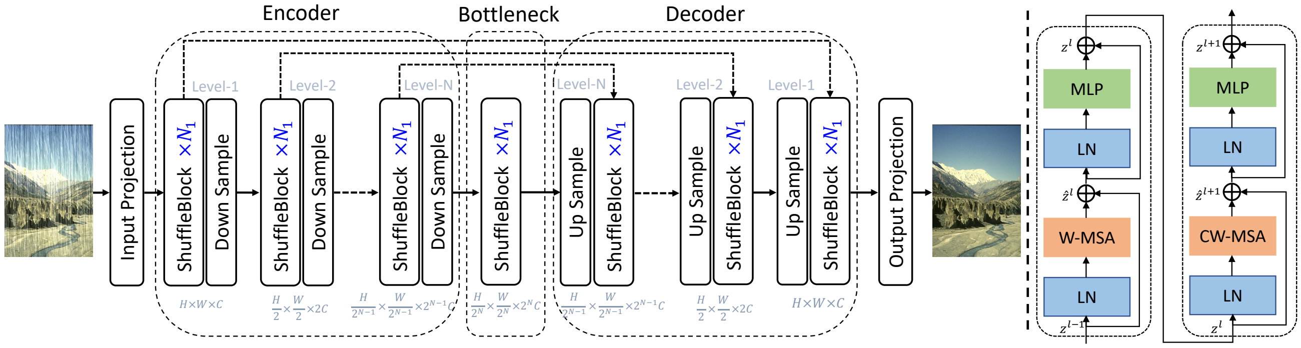
Setting. In this section, we validate the effectiveness of the proposed ShuffleFormer. Following the experimental setting of previous work (Xiao et al., 2022), we also elaborate four ShuffleFormer variants: ShuffleFormer-`SS`, ShuffleFormer-`SM`, ShuffleFormer-`CS`, ShuffleFormer-`CM`. The first symbol aims to indicate whether the chaotic window self-attention is adopted for training(`S`: the shifted window self-attention; `C`: the chaotic window self-attention) and the second symbol represents whether the Monte-Carlo averaging is adopted for testing(`S`: the shifted window self-attention; `M`: the Monte-Carlo averaging for the chaotic window self-attention). These variants are identical except for the shifted/chaotic window self-attention. In particular, ShuffleFormer-`SS` is degraded to the traditional shifted window transformer. The number of samples used in Monte-Carlo averaging is 16, *i.e.*, $M = 16$. For ShuffleFormer-`SM` and ShuffleFormer-`CM`, we should evaluate the trained model five times and report the mean as well as standard deviation in the form of mean$\pm$std. In practice, due to Monte-Carlo averaging, the standard deviation is an order of magnitude smaller compared with the precision of the mean value. Therefore, we only report the mean results for simplicity.

4.2. Image Denoising

We perform the real noise removal experiment on SIDD (Abdelhamed et al., 2018) datasets. To demonstrate the efficacy of our method, we include nine representative methods for comparison. Table 3 reports PSNR and SSIM scores of previous methods as well as ShuffleFormer variants for real-world image denoising, respectively. Figure 10 shows visual comparison with other methods.

4.3. Image Deraining

Image deraining experiments are performed on the real-world SPA-Data (Purohit et al., 2021), which 638K training pairs and 1000 testing images. Except for the elaborate four ShuffleFormer variants, existing six deraining methods are included: GMM (Li et al., 2016), DDN (Fu



(a) Architecture

(b) ShuffleBlock

Figure 4: (a) The overall architecture of ShuffleFormer; (b) the structure of ShuffleBlock.

Table 3: Quantitative results of image denoising on SIDD.

Method	SIDD	
	PSNR ↑	SSIM ↑
BM3D (Dabov et al., 2007b)	25.65	0.685
RIDNet (Anwar & Barnes, 2019)	38.71	0.914
VDN (Yue et al., 2019)	39.28	0.909
DANet (Yue et al., 2020)	39.47	0.918
CycleISP (Zamir et al., 2020a)	39.52	0.957
MIRNet (Zamir et al., 2020b)	39.72	0.959
MPRNet (Zamir et al., 2021)	39.71	0.958
NBNet (Cheng et al., 2021)	39.75	0.959
MAXIM (Tu et al., 2022)	39.96	0.960
ShuffleFormer-SS(Wang et al., 2022)	39.89	0.960
ShuffleFormer-SM	39.35	0.956
ShuffleFormer-CS	39.60	0.958
ShuffleFormer-CM	40.00	0.960

et al., 2017), SPANet (Wang et al., 2019), JORDER-E (Yang et al., 2020), RCDNet (Wang et al., 2020), SPAIR (Purohit et al., 2021). Table 4 shows the performance comparison. Figure 11 shows in comparison with other methods, ShuffleFormer-CM is more effective in removing rainy artifacts while preserving image textures.

4.4. Image Deblurring

We conduct deblurring experiments on four benchmark datasets, including two synthesized datasets (GoPro (Nah et al., 2017) and HIDE (Shen et al., 2019)), and two real-world datasets (RealBlur-R (Rim et al., 2020) and RealBlur-J (Rim et al., 2020)). Following previous works (Kupyn et al., 2019), we train ShuffleFormer only on the GoPro dataset and directly evaluate the well-trained model on GoPro, HIDE, RealBlur-R and RealBlur-J. Table 5 presents PSNR and SSIM scores of different deblurring methods and ShuffleFormer variants. Figure 12 presents an image deblurring example from GoPro (Nah et al., 2017).

Table 4: Quantitative results of image deraining on SPA-Data dataset.

Method	SPA-Data	
	PSNR ↑	SSIM ↑
GMM (Li et al., 2016)	34.30	0.9428
DDN (Fu et al., 2017)	36.97	0.9604
SPANet (Wang et al., 2019)	40.24	0.9811
JORDER-E (Yang et al., 2020)	40.78	0.9811
RCDNet (Wang et al., 2020)	41.47	0.9834
SPAR (Purohit et al., 2021)	44.10	0.9872
ShuffleFormer-SS (Wang et al., 2022)	48.80	0.9935
ShuffleFormer-SM	47.91	0.9927
ShuffleFormer-CS	48.08	0.9923
ShuffleFormer-CM	49.19	0.9935

Findings. Based on above experimental results on various image restoration tasks, we can make the following observations and analyses:

- ShuffleFormer-SS vs. ShuffleFormer-SM: Without the random shuffle for training, directly applying Monte-Carlo averaging will degrade performance dramatically, which reveals that the performance gain cannot simply attribute to feature ensemble at testing time and training with random shuffle matters;
- ShuffleFormer-CM vs. ShuffleFormer-SM: Random shuffle for training plays a vital role in performance improvements when testing with MC averaging;
- ShuffleFormer-CM vs. ShuffleFormer-CS: If only *shifted window strategy* is adopted for testing, ShuffleFormer suffers from severe performance degradation. This result seems weird since shifted window can be a particular instance of random shuffle. The distinction is that random shuffle enables window self-attention to model non-local interactions while shifted window strategy is still restricted to the local scope.

Table 5: Quantitative results of image deblurring on GoPro.

Method	GoPro		HIDE		RealBlur-R		RealBlur-J		Average	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
Xu et al. (Xu et al., 2013)	21.00	0.741	-	-	34.46	0.937	27.14	0.830	-	-
Nah et al. (Nah et al., 2017)	29.08	0.914	25.73	0.874	32.51	0.841	27.87	0.827	28.80	0.864
DeblurGAN (Kupyn et al., 2018)	28.70	0.858	24.51	0.871	33.79	0.903	27.97	0.834	28.74	0.867
Zhang et al. (Zhang et al., 2018a)	29.19	0.931	-	-	35.48	0.947	27.80	0.847	-	-
SRN (Tao et al., 2018)	30.26	0.934	28.36	0.915	35.66	0.947	28.56	0.867	30.71	0.915
DeblurGAN-v2 (Kupyn et al., 2019)	29.55	0.934	26.61	0.875	35.26	0.944	28.70	0.866	30.03	0.905
DMPHN (Zhang et al., 2019a)	31.20	0.940	29.09	0.924	35.70	0.948	28.42	0.860	31.10	0.918
DBGAN (Zhang et al., 2020)	31.10	0.942	28.94	0.915	-	-	-	-	-	-
MPRNet (Zamir et al., 2021)	32.66	0.959	30.96	0.939	35.99	0.952	28.70	0.873	32.08	0.931
DGUNet$^+$ (Mou et al., 2022)	33.17	0.963	31.40	0.944	-	-	-	-	-	-
MAXIM (Tu et al., 2022)	32.86	0.961	32.83	0.956	35.78	0.947	28.83	0.875	32.58	0.935
Restormer (Zamir et al., 2022)	32.92	0.961	31.22	0.942	36.19	0.957	28.96	0.879	32.32	0.935
ShuffleFormer-SS (Wang et al., 2022)	33.05	0.962	30.89	0.940	36.19	0.956	29.06	0.884	32.30	0.936
ShuffleFormer-SM	31.74	0.953	29.77	0.928	35.40	0.944	28.01	0.852	31.23	0.919
ShuffleFormer-CS	32.38	0.958	30.42	0.936	35.75	0.951	28.40	0.866	31.74	0.928
ShuffleFormer-CM	33.38	0.965	31.25	0.943	36.34	0.958	29.19	0.890	32.54	0.939

4.5. Analytic Experiment

Trade-off of running time and performance. We analyze the trade-off of running time and performance brought by Monte-Carlo averaging. We conduct experiment on SIDD dataset (Abdelhamed et al., 2018) and evaluate the trend of running time and PSNR as the number of MC samples increases. Each experiment is repeated 5 times and then compute the mean and standard deviation. Figure 5 presents the results. We can draw the conclusion: i) Even when the number of MC samples is 16, it only incurs approximate $4\times$ additional running time (rather than $16\times$); ii) We can still obtain significant performance improvements in the case of a single sample where the efficiency is the same with the shifted window self-attention. The trend of memory footprint against Monte-Carlo samples can be found in Appendix D.1.

Effect of position modeling. To make use of positional information, we employ a single depth-wise convolution following (Chu et al., 2023). We further conduct experiments for validate the effectiveness of a depth-wise convolution for position modeling on SIDD dataset and the results are reported in Table 6. It can be found that the depth-wise convolution can model relative position based interactions to slightly promote the final performance.

Visualization of attention map. To further validate that the capacity of modeling non-local interactions, we provide visualization of the attention map in Figure 9. We can observe that attention map of SW-MSA is restricted within a local window. In contrast, CW-MSA with random shuffle breaks the local barrier brought by window partition and capture non-local interactions. Please refer to Appendix E.1.2 for more details.

Table 6: Abalation study about position modeling based on SIDD dataset.

Method	PSNR (dB) ↑	SSIM ↑
w/o Dconv	39.93	0.960
+ Dconv	40.00	0.960

Consistent improvements across window size. In Assumption 3.1, we hypothesize that CW-MSA utilizes local window self-attention whose size is much smaller compared with the input resolution to model non-local interactions. We here further demonstrate that CW-MSA can still attain consistent improvements across different window size. We perform real-world denoising experiments based on SIDD dataset (Abdelhamed et al., 2018) and choose three different window size $\{2\times 2, 4\times 4, 8\times 8\}$, which still respects the local partition in Assumption 3.1. Figure 5 shows that CW-MSA exhibits consistent superiority to the shifted window baseline across varying small window size, which is attributed to its significantly increased modeling distance by the random shuffle transformation pair.

5. Limitation

We provide a fresh perspective to increase the modeling distance of window self-attention by randomly shuffling the input. It should be emphasized that the basics of applying random shuffle to self-attention is the permutation-equivariance property. In addition, we have also extended the random shuffle strategy to CNN and the experiments reveal that CNN can not easily benefit from the extended receptive field by random shuffle. Please refer to Appendix D.2 for more evidence. As shown in Figure 5, random shuffle when integrated with CNN leads to a severe oscillation in perfor-

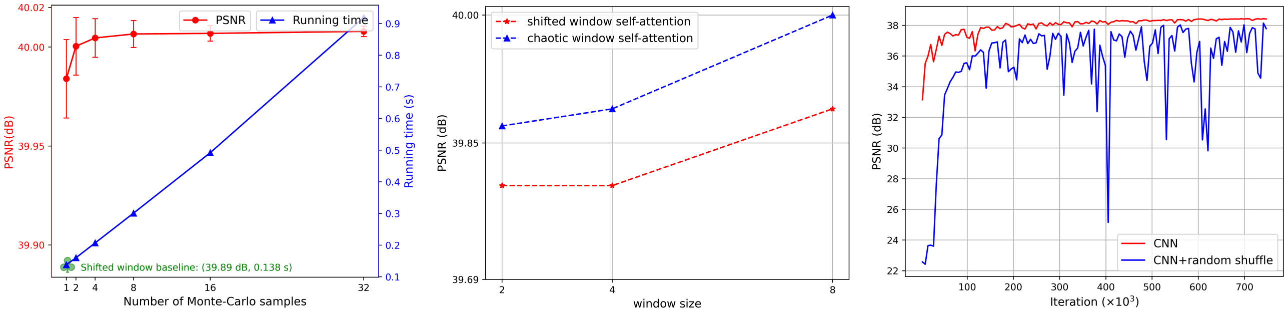


Figure 5: *Left*: Trade-off between performance and running time. *Middle*: Consistent improvements over varying window size. *Right*: CNN cannot benefit from random shuffle.

mance. All observations lead to a conclusion that random shuffle impairs the modeling ability of CNN. The reason is that CNN depends its modeling solely on the relative position, which is no longer reliable after random shuffle. Exploring the way to adapt the random shuffle to CNN or constructing a completely new transformation to extend the receptive field of CNN is an interesting topic for us.

6. Conclusion

In this paper, we attempt to model non-local interactions with only local window Transformer for image restoration. Given the philosophy that motion is relative, we propose to randomly shuffle the input rather than shift window strategy before passing self-attention. Random shuffle breaks the local barrier incurred by window partition and facilitates local self-attention to model non-local interactions without introducing extra parameters. The derived CW-SA can be trained efficiently by sampling the random shuffle. For testing, we elaborate the Monte-Carlo averaging to approximate the expectation of the introduced random factors. Our method enjoys both elegant theoretical guarantees and superior performance for classic image restoration tasks.

Acknowledgement

This work was supported by the National Key R&D Program of China under Grant 2020AAA0105702, the National Natural Science Foundation of China (NSFC) under Grants 62225207, U19B2038, 62121002 and 62276243.

References

Abdelhamed, A., Lin, S., and Brown, M. S. A high-quality denoising dataset for smartphone cameras. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

Anwar, S. and Barnes, N. Real image denoising with feature attention. In *Int. Conf. Comput. Vis.*, 2019.

Batson, J. and Royer, L. Noise2self: Blind denoising by self-supervision. In *Int. Conf. Mach. Learn.*, 2019.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., and Le, Q. V. Attention augmented convolutional networks. In *Int. Conf. Comput. Vis.*, 2019.

Charbonnier, P., Blanc-Feraud, L., Aubert, G., and Barlaud, M. Two deterministic half-quadratic regularization algorithms for computed imaging. In *IEEE Int. Conf. Image Process.*, 1994.

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., and Gao, W. Pre-trained image processing transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

Cheng, S., Wang, Y., Huang, H., Liu, D., Fan, H., and Liu, S. Nbnet: Noise basis learning for image denoising with subspace projection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., and Shen, C. Conditional positional encodings for vision transformers. In *Int. Conf. Learn. Represent.*, 2023.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. On the relationship between self-attention and convolutional layers. In *Int. Conf. Learn. Represent.*, 2019.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 2007a.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 2007b.

Dai, T., Cai, J., Zhang, Y., Xia, S.-T., and Zhang, L. Second-order attention network for single image super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Dai, Z., Liu, H., Le, Q. V., and Tan, M. Coatnet: Marrying convolution and attention for all data sizes. In *Adv. Neural Inform. Process. Syst.*, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dong, J., Roth, S., and Schiele, B. Deep wiener deconvolution: Wiener meets deep learning for image deblurring. In *Adv. Neural Inform. Process. Syst.*, 2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2021.

Fan, Y., Yu, J., Mei, Y., Zhang, Y., Fu, Y., Liu, D., and Huang, T. S. Neural sparse representation for image restoration. In *Adv. Neural Inform. Process. Syst.*, 2020.

Fu, X., Huang, J., Ding, X., Liao, Y., and Paisley, J. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Trans. Image Process.*, 2017.

Haris, M., Shakhnarovich, G., and Ukita, N. Deep back-projection networks for super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

He, K., Sun, J., and Tang, X. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011.

Huang, Y., Li, S., Wang, L., Tan, T., et al. Unfolding the alternating optimization for blind super resolution. In *Adv. Neural Inform. Process. Syst.*, 2020.

Huynh-Thu, Q. and Ghanbari, M. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 2008.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., and Matas, J. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

Kupyn, O., Martyniuk, T., Wu, J., and Wang, Z. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Int. Conf. Comput. Vis.*, 2019.

Laine, S., Karras, T., Lehtinen, J., and Aila, T. High-quality self-supervised deep image denoising. In *Adv. Neural Inform. Process. Syst.*, 2019.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 2015.

Lefkimmiatis, S. Non-local color image denoising with convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., and Aila, T. Noise2noise: Learning image restoration without clean data. In *Int. Conf. Mach. Learn.*, 2018.

Li, C., Guo, C., Han, L.-H., Jiang, J., Cheng, M.-M., Gu, J., and Loy, C. C. Low-light image and video enhancement using deep learning: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021a.

Li, C., Guo, C., and Loy, C. C. Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021b.

Li, Y., Tan, R. T., Guo, X., Lu, J., and Brown, M. S. Rain streak removal using layer priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., and Timofte, R. Swinir: Image restoration using swin transformer. In *Int. Conf. Comput. Vis. Worksh.*, 2021.

Lin, J., Cai, Y., Hu, X., Wang, H., Yan, Y., Zou, X., Ding, H., Zhang, Y., Timofte, R., and Van Gool, L. Flow-guided sparse transformer for video deblurring. In *Int. Conf. Mach. Learn.*, 2022a.

Lin, J., Hu, X., Cai, Y., Wang, H., Yan, Y., Zou, X., Zhang, Y., and Van Gool, L. Unsupervised flow-aligned sequence-to-sequence learning for video restoration. In *Int. Conf. Mach. Learn.*, 2022b.

Liu, D., Wen, B., Fan, Y., Loy, C. C., and Huang, T. S. Non-local recurrent network for image restoration. In *Adv. Neural Inform. Process. Syst.*, 2018.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, 2021.

Luo, F., Wu, X., and Guo, Y. Functional neural networks for parametric image restoration problems. In *Adv. Neural Inform. Process. Syst.*, 2021.

Mou, C., Zhang, J., and Wu, Z. Dynamic attentive graph learning for image restoration. In *Int. Conf. Comput. Vis.*, 2021.

Mou, C., Wang, Q., and Zhang, J. Deep generalized unfolding networks for image restoration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

Nah, S., Hyun Kim, T., and Mu Lee, K. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

Nguyen, T. M., Nguyen, T. M., Le, D. D., Nguyen, D. K., Tran, V.-A., Baraniuk, R., Ho, N., and Osher, S. Improving transformers with probabilistic attention keys. In *Int. Conf. Mach. Learn.*, 2022.

Plötz, T. and Roth, S. Neural nearest neighbors networks. In *Adv. Neural Inform. Process. Syst.*, 2018.

Purohit, K., Suin, M., Rajagopalan, A., and Boddeti, V. N. Spatially-adaptive image restoration using distortion-guided networks. In *Int. Conf. Comput. Vis.*, 2021.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. In *Adv. Neural Inform. Process. Syst.*, 2019.

Ren, D., Zuo, W., Hu, Q., Zhu, P., and Meng, D. Progressive image deraining networks: A better and simpler baseline. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Ren, W., Zhang, J., Ma, L., Pan, J., Cao, X., Zuo, W., Liu, W., and Yang, M.-H. Deep non-blind deconvolution via generalized low-rank approximation. In *Adv. Neural Inform. Process. Syst.*, 2018.

Ren, W., Zhang, J., Pan, J., Liu, S., Ren, J., Du, J., Cao, X., and Yang, M.-H. Deblurring dynamic scenes via spatially varying recurrent neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

Rim, J., Lee, H., Won, J., and Cho, S. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *Eur. Conf. Comput. Vis.*, 2020.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

Shen, Z., Wang, W., Lu, X., Shen, J., Ling, H., Xu, T., and Shao, L. Human-aware motion deblurring. In *Int. Conf. Comput. Vis.*, 2019.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014.

Tao, X., Gao, H., Shen, X., Wang, J., and Jia, J. Scale-recurrent network for deep image deblurring. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al. Mlp-mixer: An all-mlp architecture for vision. In *Adv. Neural Inform. Process. Syst.*, 2021.

Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., and Li, Y. Maxim: Multi-axis mlp for image processing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017.

Wang, H., Xie, Q., Zhao, Q., and Meng, D. A model-driven deep neural network for single image rain removal. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., and Lau, R. W. Spatial attentive single-image deraining with a high quality real rain dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Wang, Y., Yang, J., Yin, W., and Zhang, Y. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 2008.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 2004.

Wang, Z., Cun, X., Bao, J., and Liu, J. Uformer: A general u-shaped transformer for image restoration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

Wu, H., Wu, J., Xu, J., Wang, J., and Long, M. Flowformer: Linearizing transformers with conservation flows. In *Int. Conf. Mach. Learn.*, 2022.

Xiao, J., Fu, X., Wu, F., and Zha, Z.-J. Stochastic window transformer for image restoration. In *Adv. Neural Inform. Process. Syst.*, 2022.

Xu, L., Zheng, S., and Jia, J. Unnatural l0 sparse representation for natural image deblurring. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013.

Yang, W., Tan, R. T., Feng, J., Liu, J., Guo, Z., and Yan, S. Deep joint rain detection and removal from a single image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

Yang, W., Tan, R. T., Feng, J., Guo, Z., Yan, S., and Liu, J. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

Yue, Z., Yong, H., Zhao, Q., Meng, D., and Zhang, L. Variational denoising network: Toward blind noise modeling and removal. In *Adv. Neural Inform. Process. Syst.*, 2019.

Yue, Z., Zhao, Q., Zhang, L., and Meng, D. Dual adversarial network: Toward real-world noise removal and noise generation. In *Eur. Conf. Comput. Vis.*, 2020.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., and Shao, L. Cycleisp: Real image restoration via improved data synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020a.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., and Shao, L. Learning enriched features for real image restoration and enhancement. In *Eur. Conf. Comput. Vis.*, 2020b.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., and Shao, L. Multi-stage progressive image restoration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., and Yang, M.-H. Restormer: Efficient transformer for high-resolution image restoration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

Zhang, H., Dai, Y., Li, H., and Koniusz, P. Deep stacked hierarchical multi-patch network for image deblurring. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019a.

Zhang, H., Sindagi, V., and Patel, V. M. Image de-raining using a conditional generative adversarial network. *IEEE Trans. Circuit Syst. Video Technol.*, 2019b.

Zhang, J., Pan, J., Ren, J., Song, Y., Bao, L., Lau, R. W., and Yang, M.-H. Dynamic scene deblurring using spatially variant recurrent neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018a.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.*, 2017a.

Zhang, K., Zuo, W., Gu, S., and Zhang, L. Learning deep cnn denoiser prior for image restoration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017b.

Zhang, K., Zuo, W., and Zhang, L. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Trans. Image Process.*, 2018b.

Zhang, K., Luo, W., Zhong, Y., Ma, L., Stenger, B., Liu, W., and Li, H. Deblurring by realistic blurring. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

Zhang, Y., Li, K., Li, K., Zhong, B., and Fu, Y. Residual non-local attention networks for image restoration. In *Int. Conf. Learn. Represent.*, 2019c.

Zhang, Y., Li, K., Li, K., Sun, G., Kong, Y., and Fu, Y. Accurate and fast image denoising via attention guided scaling. *IEEE Trans. Image Process.*, 2021a.

Zhang, Y., Wang, H., Qin, C., and Fu, Y. Aligned structured sparsity learning for efficient image super-resolution. In *Adv. Neural Inform. Process. Syst.*, 2021b.

A. Proof to Theorem Theorem 3.5

Proof. Each pixel has the equal probability to be in the same window with pixel $\mathbf{m}_1 = (m_{h_1}, m_{w_1})$. Hence, the expected chaotic distance is computed as:

$$\begin{aligned} d(\mathbf{m}_1) &= \mathop{\mathbb{E}}_{\mathbf{m}_2|(\mathbf{m}_1,\mathbf{m}_2)\in\mathcal{C}_\mathcal{S}} [d_\mathcal{S}(\mathbf{m}_1, \mathbf{m}_2)] \\ &= \frac{1}{HW-1} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \sqrt{(h - m_{h_1})^2 + (w - m_{w_1})^2}. \end{aligned} \tag{14}$$

Given the mean inequality

$$\sqrt{\frac{x^2+y^2}{2}} \geq \frac{x+y}{2}, \tag{15}$$

$d(\mathbf{m}_1)$ has the lower bound:

$$\begin{aligned} d(\mathbf{m}_1) &= \frac{1}{HW-1} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \sqrt{(h - m_{h_1})^2 + (w - m_{w_1})^2} \\ &\geq \frac{1}{HW-1} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \frac{\sqrt{2}}{2} \left(|h - m_{h_1}| + |w - m_{w_1}|\right). \end{aligned} \tag{16}$$

Note that

$$\begin{aligned} &\sum_{w=0}^{W-1} |w - m_{w_1}| \\ =& \sum_{w=0}^{m_{w_1}} (m_{w_1} - w) + \sum_{w=m_{w_1}}^{W-1} (w - m_{w_1}) \\ =& \frac{m_{w_1}(m_{w_1}+1)}{2} + \frac{(W - m_{w_1})(W - m_{w_1} - 1)}{2} \end{aligned} \tag{17}$$

and

$$\begin{aligned} &\sum_{h=0}^{H-1} |h - m_{h_1}| \\ =& \sum_{h=0}^{m_{h_1}} (m_{h_1} - h) + \sum_{h=m_{h_1}}^{H-1} (h - m_{h_1}) \\ =& \frac{m_{h_1}(m_{h_1}+1)}{2} + \frac{(H - m_{h_1})(H - m_{h_1} - 1)}{2}, \end{aligned} \tag{18}$$

we then have:

$$\begin{aligned} d(\mathbf{m}_1) \geq & \frac{\sqrt{2}}{4(HW-1)} [H m_{w_1}(m_{w_1}+1) + W m_{h_1}(m_{h_1}+1) \\ & + H(W - m_{w_1})(W - m_{w_1} - 1) + W(H - m_{h_1})(H - m_{h_1} - 1)] \end{aligned} \tag{19}$$

which completes the proof. □

B. Experimental Setting

Loss function. The loss function adopted for training is the Charbonnier loss (Charbonnier et al., 1994), whose mathematical formula is:

$$L(I', I) = \sqrt{||I' - I||^2 + \epsilon^2}, \tag{20}$$

where I' and I are the restored and ground-truth image respectively. The constant ϵ is empirically set to 10^{-3}.

Training Detail. Following Uformer (Wang et al., 2022), ShuffleFormers employ a four-level encoder-decoder structure. The numbers of ShuffleBlock are $\{1, 2, 8, 8\}$ for level-1 to level-4 of Encoder and the blocks for Decoder are mirrored. The number of channel is set to 32 and the window size is 8×8. We train the network with Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) with the initial learning rate 2×10^{-4} gradually reduced to 1×10^{-6} with the cosine annealing. The training samples are augmented by the horizontal flipping and rotation of 90°, 180°, or 270°. We then describe the task-specific settings.

Image deraining. We train ShuffleFormers using four TITAN Xp GPUs with batch size 8 on 256×256 image pairs. The training process lasts for 10 epochs. Following previous works (Wang et al., 2020; Yang et al., 2017), We evaluate PSNR (Huynh-Thu & Ghanbari, 2008) and SSIM (Wang et al., 2004) based on the luminance channel, i.e., Y channel of YCbCr space.

Image denosing. For image denoising, we conduct real noise removal experiment on SIDD (Abdelhamed et al., 2018) dataset. The training patches are cropped from the total training set with size 128×128. We train ShuffleFormers using four TITAN Xp GPUs for total 250 epoches with batch size 32 and PSNR is evaluated on the full-size test images.

Image deblurring. ShuffleFormers are trained on GoPro dataset (Nah et al., 2017)[2] and directly applied to GoPro (Nah et al., 2017), HIDE (Shen et al., 2019), RealBlur-R (Rim et al., 2020) and RealBlur-J (Rim et al., 2020). We crop 512×512 image patches with stride 256 from GoPro dataset and train ShuffleFormers with 256×256 training pairs randomly cropped from 512×512 image patches. The total training epoch is 600 with batch size 8 on four TITAN Xp GPUs and we evaluate PSNR and SSIM on the full-size test images.

C. Implementation for random shuffle and inverse shuffle

We provide python implementation for the random shuffle and corresponding inverse shuffle in Algorithm 1 and 2, which only involves rearrangement operations.

Algorithm 1 Python Implementation of random shuffle

```
import random
def random_shuffle(x):
    B, H, W, C = x.shape
    H_shuffle = random.shuffle(list(range(0, H))) #shuffle index
    W_shuffle = random.shuffle(list(range(0, W)))
    shuffle_x = x[:, H_Shuffle, :, :] #shuffle on H
    shuffle_x = shuffle_x[:, :, W_Shuffle, :]#shuffle on W
    return shuffle_x, (H_shuffle, W_shuffle)
```

Algorithm 2 Python Implementation of inverse shuffle

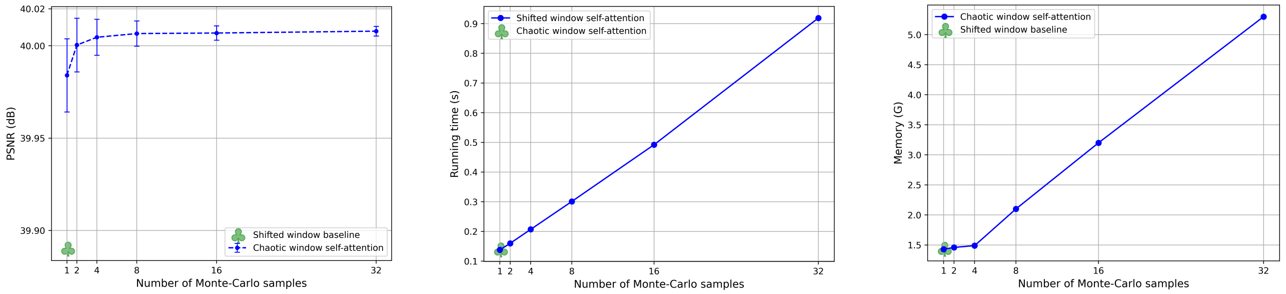
```
def inverse_shuffle(x, (H_shuffle, W_shuffle)):
    B, H, W, C = x.shape
    H_Index, W_Index = list(range(0, H)), list(range(0, W))
    inverse_x[:, :, W_Shuffle, :] = x[:, :, W_Index, :] #inverse for W
    inverse_x[:, H_Shuffle, :, :] = inverse_x[:, H_Index, :, :] #inverse for H
    return inverse_x
```

D. Analytic Experiments

D.1. Trade-off of resource and performance

Chaotic window self-attention allows to trade resource including computation and memory for performance. To validate, we conduct experiments on SIDD (Abdelhamed et al., 2018) and evaluate the trend of resource consumption and PSNR as the number of samples increases. Every experiment is repeated 5 times and then compute the mean and standard deviation. Figure 6 shows the results. We can draw the following conclusions: i) the performance and consumed resources increase with the number of MC samples, which provides a performance-efficiency trade-off; ii) Even when the number of MC samples is 16, it only incurs approximate 4× (rather than 16×) additional running time (Figure 6(b)) and memory

[2] https://seungjunnah.github.io/Datasets/gopro, CC BY 4.0 license.



(a) PSNR vs. # MC sample

(b) Running time vs. # MC sample

(c) Memory vs. # MC sample

Figure 6: Monte-Carlo averaging can trade resource for accurate restoration results.

consumption (Figure 6(c)); iii) in the case of a single sample where the efficiency is the same with tranditional shifted window self-attention, we can still obtain significant performance improvements.

Discussion: why is Monte-Carlo averaging needed given the fact that we can get promising results when the number of MC sample is 1? We design the algorithm based on two considerations: i) To ensure theoretical completeness of our random shuffle framework. Since random shuffle introduces random factors in Transformer, we must propose a way to approximate the model expectation as Dropout does; ii) Monte-Carlo averaging can reduce the variance of restored results (see Figure 6(a)); iii) we provide a theoretical interpretation that Monte-Carlo averaging boosts the model performance: suppose that the model is denoted by F, the random factors are collectively denoted $\mathcal{S}$, we utilize the square of L_2 norm as the criterion to evaluate the fitted network. Given the degraded image x, the expected loss of the fitted $F(x;\mathcal{S})$ is given by

$$\begin{aligned}\mathbb{E}_{\mathcal{S}}[||F(x,\mathcal{S})-I(x)||_2^2] &= \mathbb{E}_{\mathcal{S}}[||F(x,\mathcal{S})-\mathbb{E}_{\mathcal{S}}[F(x,\mathcal{S})]+\mathbb{E}_{\mathcal{S}}[F(x,\mathcal{S})]-I(x)||_2^2] &(21)\\ &= \mathbb{E}_{\mathcal{S}}[||F(x,\mathcal{S})-\mathbb{E}_{\mathcal{S}}[F(x,\mathcal{S})||_2^2]+||\,\mathbb{E}_{\mathcal{S}}[F(x,\mathcal{S})]-I(x)||_2^2 &(22)\\ &\geq ||\,\mathbb{E}_{\mathcal{S}}[F(x,\mathcal{S})]-I(x)||_2^2. &(23)\end{aligned}$$

Therefore, taking expectation over random factors ($\mathbb{E}_{\mathcal{S}}[F(x,\mathcal{S})]$) can boost performance against single forward process ($F(x,\mathcal{S})$) from the sense of expected loss. The true marginalization involves prohibitively expensive computational burden (since the combinations of random factors are exponentially huge). We take two steps of approximations to derive our final layer-wise Monte-Carlo averaing algorithm from the true model expectation. First, in the similar spirit with testing strategy of Dropout (Srivastava et al., 2014), we adopt layer-wise expectation to approximate the expensive model average. Second, for a certain chaotic attention layer, we adopt the Monte-Carlo averaging to approximate the exact layer-wise expectation.

D.2. Extending random shuffle to CNN

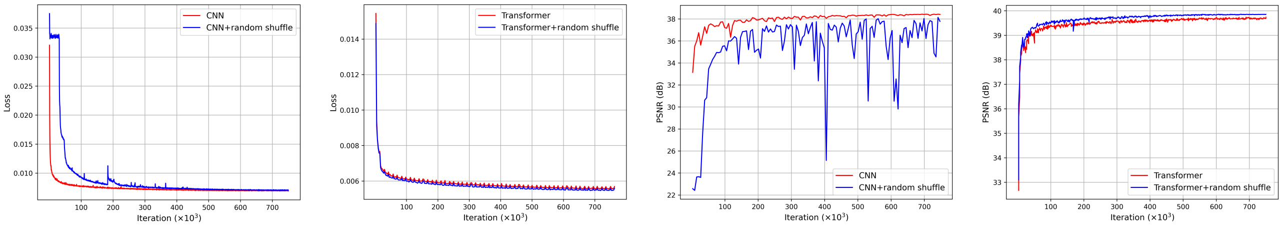
Again with the philosophy of local→non-local modeling, we extend random shuffle to CNN by alternating regular convolution and 'chaotic' convolution. We adopt the classic DnCNN (Zhang et al., 2017a) as the basic model and modify it to the chaotic version by absorbing random shuffle into half its convolution layers. Besides, we also include Shuffleformer with window size 4 for comparison (Figure 7(b) and Figure 7(d)). Experiments are conducted on SIDD dataset (Abdelhamed et al., 2018). Note that random shuffle does not introduce changes in model parameters or FLOPs. Figure 7 shows the loss curve during training and PSNR curve on testing set. Figure 7(a) shows that random shuffle imposes difficulty on CNN. In contrast, when equipped with transformer, random shuffle even facilitates optimization as illustrates in Figure 7(b). Figure 7(c) reveals that random shuffle leads to unstable and worse performance of CNN on unseen data, which is distinct from Transformer (Figure 7(d)). These observations conform to our argument that the key of introducing random shuffle to Transformer is the permutation-equivariance property of self-attention.

E. Visualization

E.1. Non-Local Interactions

E.1.1. Lower Bound of Chaotic Distance

We examine the ability of modeling non-local interactions by visualizing the lower bound of the expected chaotic distance $d(\mathbf{m}_1)$ in Figure 8. Specifically, we consider two representative instances of $\mathbf{m}_1=(0,0)$ (the red dot in Figure 8(a)) and



(a) Training loss of CNN (b) Training loss of Transformer (c) PSNR of CNN (d) PSNR of Transformer

Figure 7: Extension of random shuffle to CNN.

$\mathbf{m}_1 = (\frac{H}{2}, \frac{W}{2})$ (the red dot in Figure 8(b)). Based on Theorem 3.5, we obtain the expressions for these two instances:

$$\begin{aligned} d\left((0,0)\right) &\geq \frac{\sqrt{2}HW(H+W-2)}{4(HW-1)}, \\ d\left(\left(\frac{H}{2}, \frac{W}{2}\right)\right) &\geq \frac{\sqrt{2}HW(H+W)}{8(HW-1)}. \end{aligned} \tag{24}$$

Figure 8 presents the lower bound of modeling distance using a red circle whose radius is proportional to the distance. From Assumption 3.1, the size of local window used to perform self-attention is much smaller than the input resolution [3]. However, random shuffle can break the local barrier of local window. As shown in Figure 8, the lower bound of expected distance of interactions captured by chaotic window self-attention is comparable to the input resolution.

E.1.2. Attention Map

We further validate that the capacity of modeling non-local interactions by visualizing attention map. To mitigate the dependency on input content, we randomly pick 10 test images from SIDD and then record averaged attention maps. Attention maps from second encoding block are considered. We consider the case that the query pixel, which is marked by white point in the left column of Figure 9, locates in the centre of the feature map. As shown in the middle column of Figure 9, attention map of SW-MSA is restricted within a local window, which hinders to capture non-local interactions. In contrast, with random shuffle, CW-MSA breaks the local barrier of window partition. Figure 9 reveals that CW-MSA is capable of aggregating information from the non-local region. It is noteworthy that CW-MSA (right column of Figure 9) contains more attended pixels compared to the case of SW-MSA (middle column of Figure 9). It is because Monte-Carlo averaging ($M = 16$) is adopted for the testing case of CW-MSA.

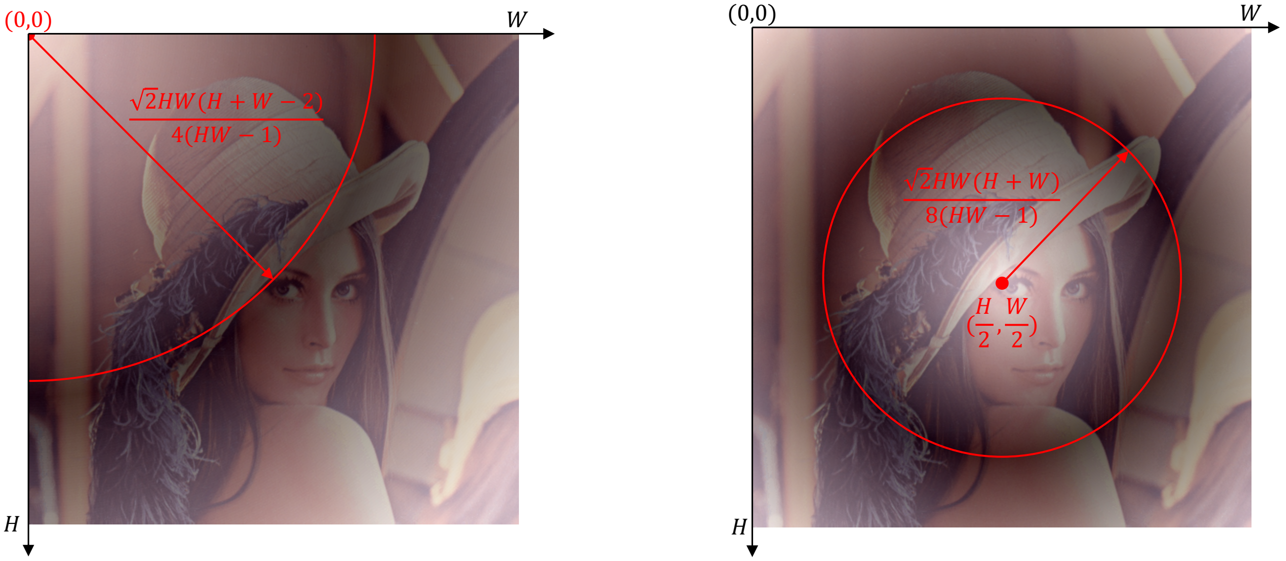
E.2. Visualization of Image Restoration

We also provide more visual results on image denoising (Figure 10), image deraining (Figure 11), and image deblurring (Figure 12). In comparison with other state-of-the-art methods and other ShuffleFormer variants, ShuffleForme-`CM`, which is equipped with random shuffle strategy for training and Monte-Carlo averaging for testing, can recover more image textures and further generate visually faithful results.

Broader Impacts

Generally, image acquisition system tends to suffer from various degenerations, including inherent noise of capturing instruments, shaking during shooting, unpredictable weather condition, and so forth. Therefore, image restoration has practical value in research and application. Our proposed random shuffle strategy can extend the modeling distance of the window transformer so that improve the performance on several tasks. Nevertheless, some negative consequences may come along. For instance, the deviation from the actual image textures caused by image restoration technologies may lead to unfair judgments in medical and criminal situations. In these scenarios, it is required to consult with human experts to avoid misjudgments.

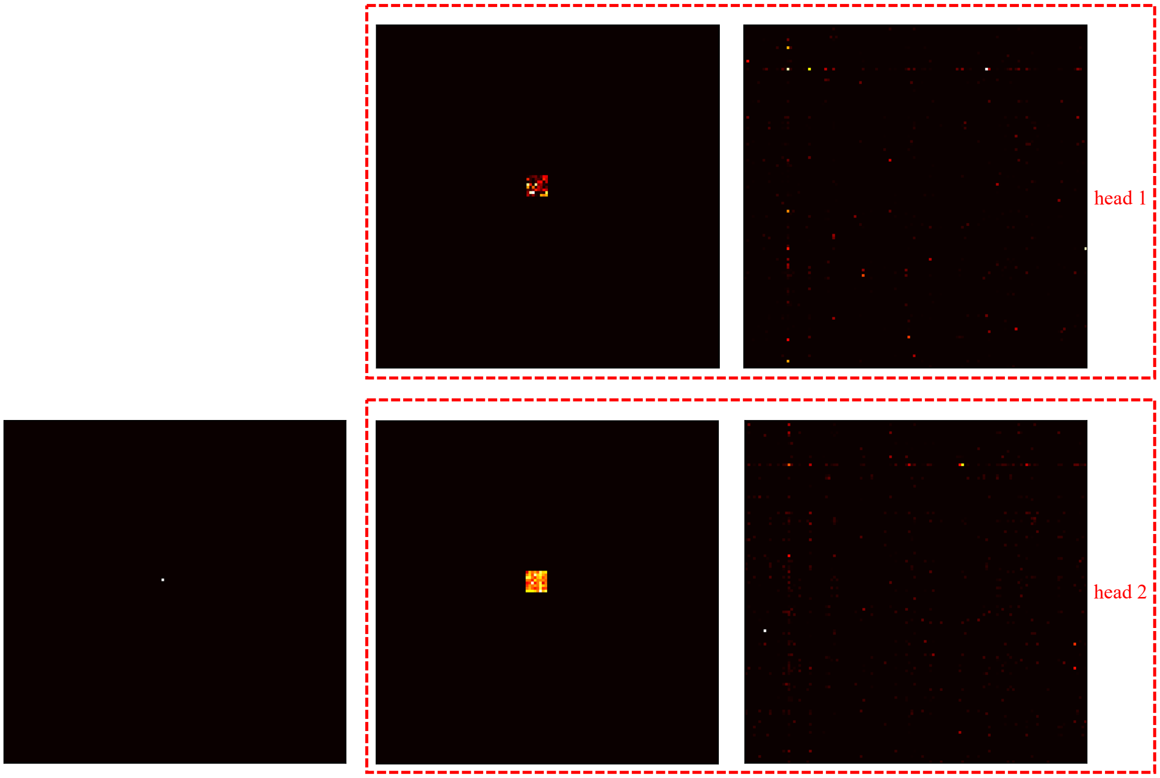
[3] In practice, the window size is 8×8 while the input resolution is 512×512.


(0,0)
W
H
$\frac{\sqrt{2}HW(H+W-2)}{4(HW-1)}$
(0,0)
W
H
$\frac{\sqrt{2}HW(H+W)}{8(HW-1)}$
$(\frac{H}{2},\frac{W}{2})$

(a) $\mathbf{m}_1 = (0, 0)$

(b) $\mathbf{m}_1 = (\frac{H}{2}, \frac{W}{2})$

Figure 8: Visualization of the lower bound of the expected chaotic distance $d(\mathbf{m}_1)$. The red dot is the reference pixel $\mathbf{m}_1$. The radius of red circle depicts the lower bound of the expected chaotic distance.

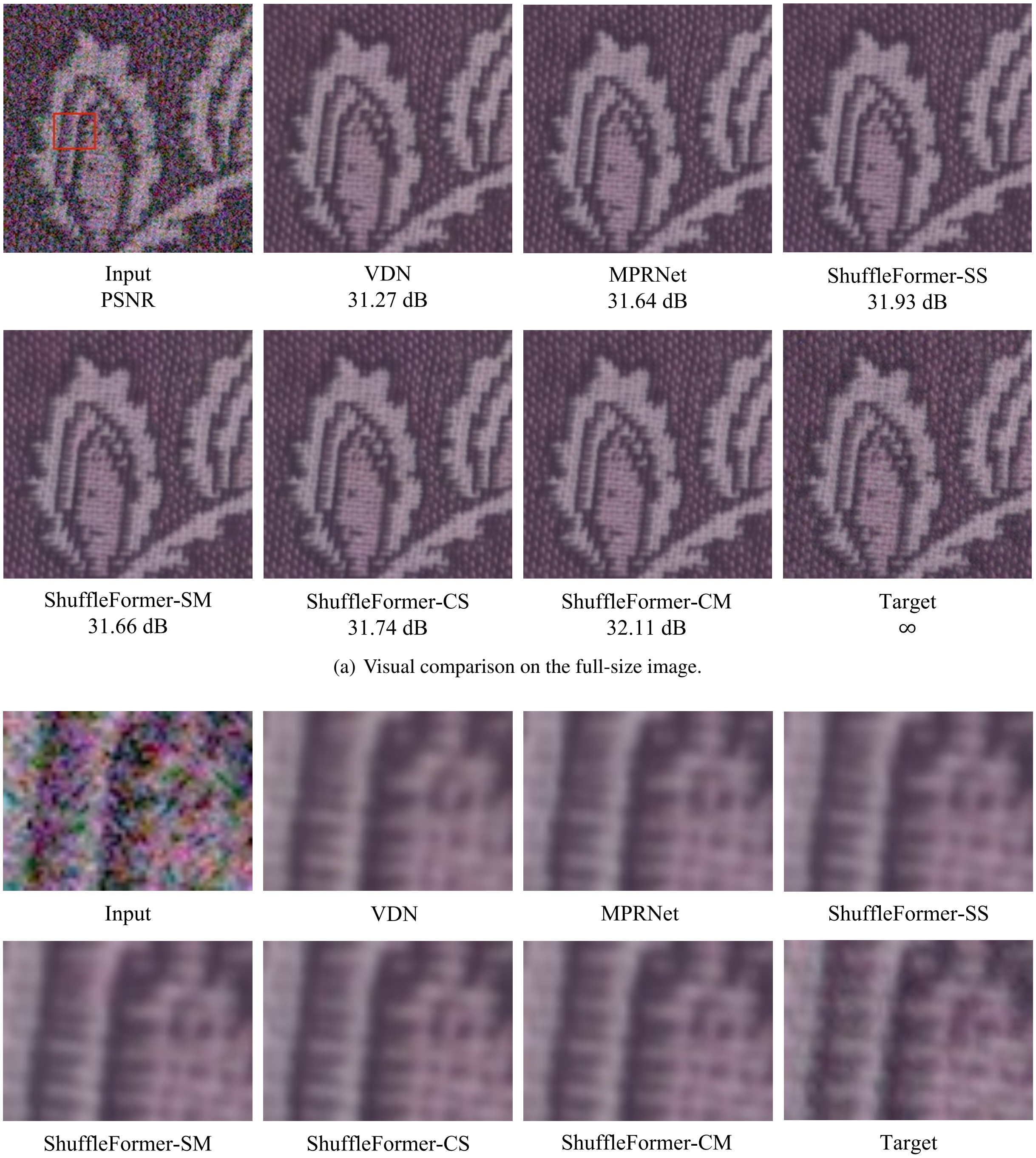


(a) The query

(b) Attention map of SW-MSA

(c) Attention map of CW-MSA

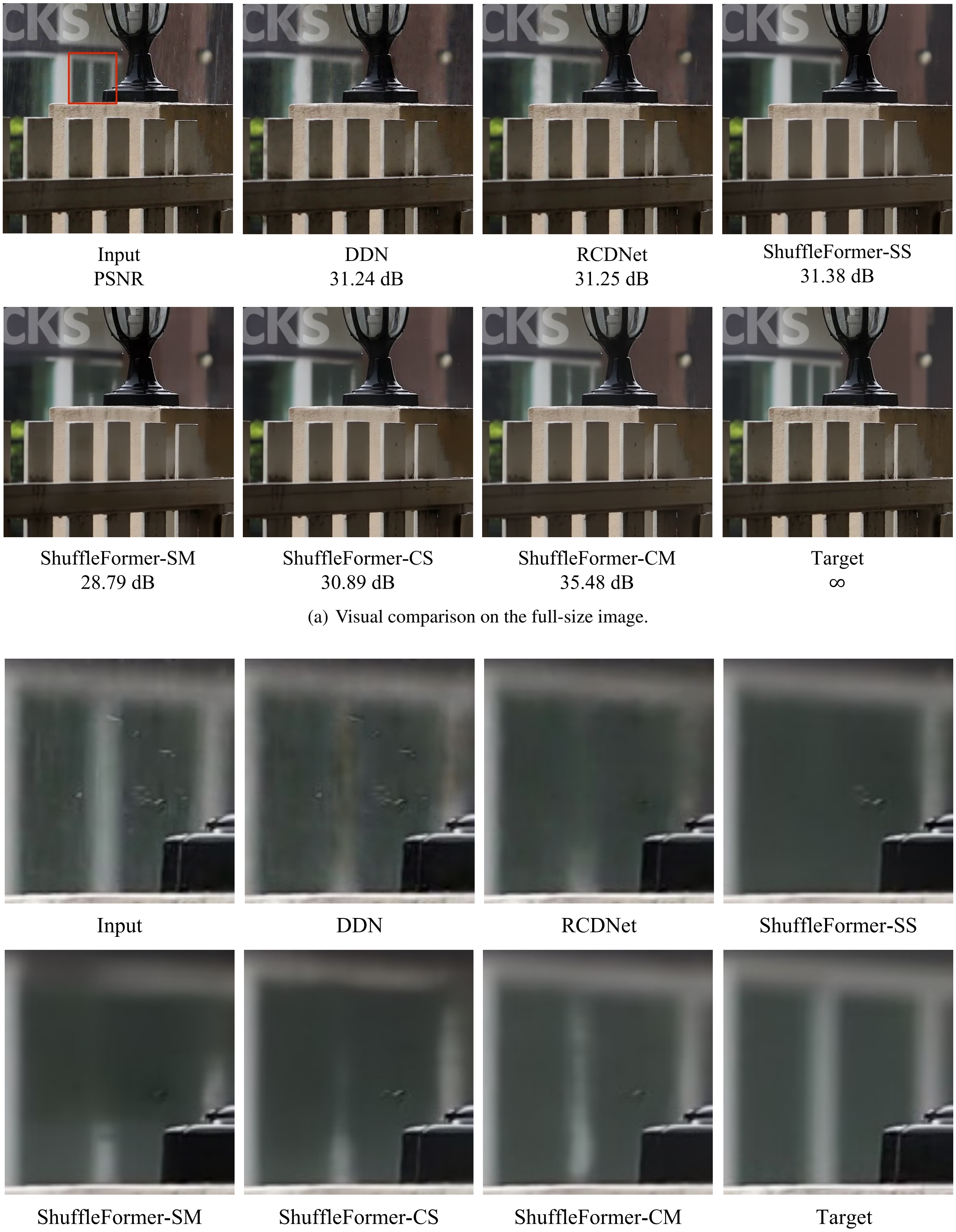
Figure 9: Visual comparison of attention map of SW-MSA and CW-MSA.



(a) Visual comparison on the full-size image.

(b) Enlarged region of Figure 10(a).

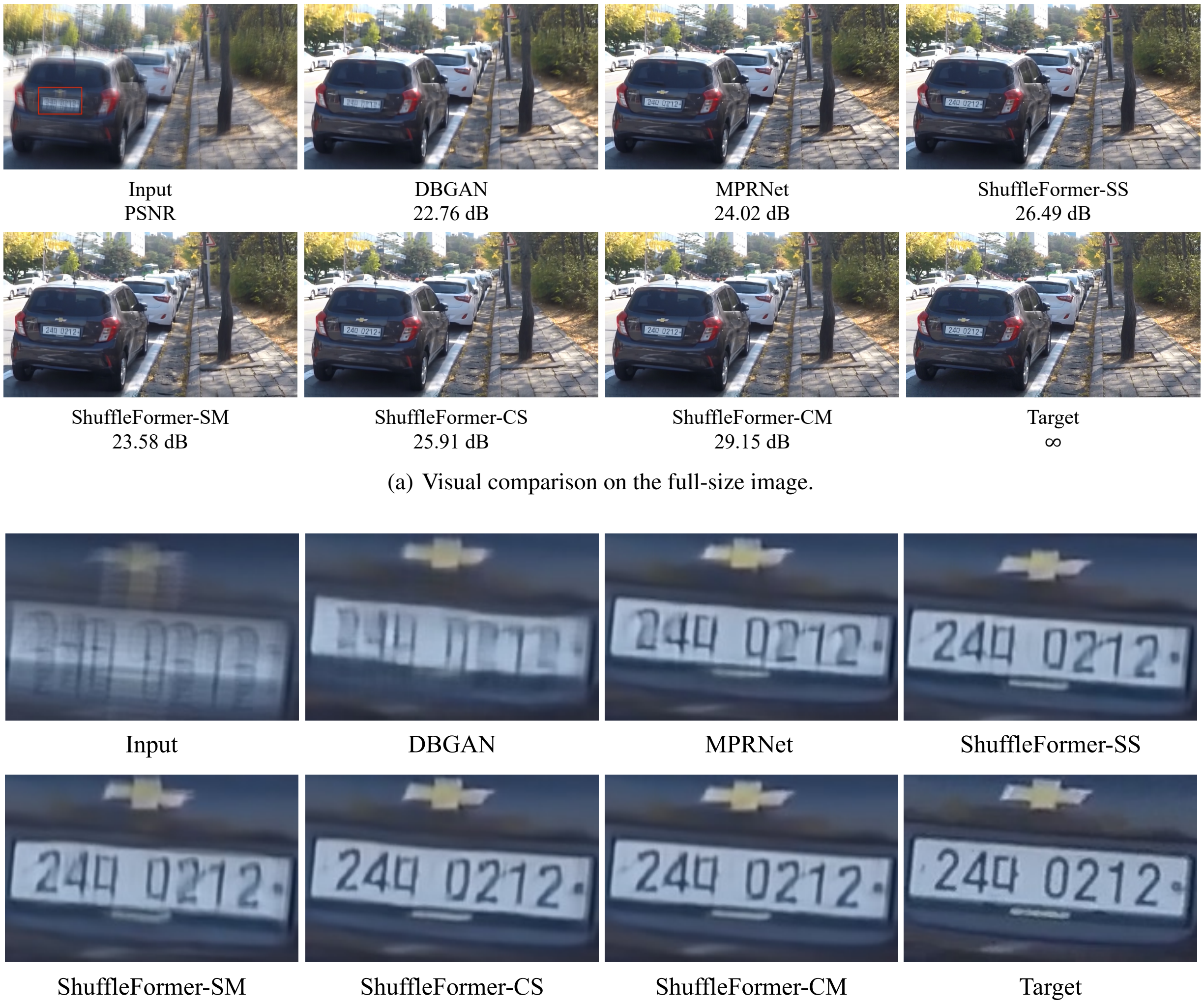
Figure 10: Visualization of image denoising on SIDD.



(a) Visual comparison on the full-size image.

(b) Enlarged region of Figure 11(a).

Figure 11: Visualization of image deraining on SPA-Data.



(a) Visual comparison on the full-size image.

(b) Enlarged region of Figure 12(a).

Figure 12: Visualization of image deblurring on GoPro.