Deep Idempotent Network for Efficient Single Image Blind Deblurring

Yuxin Mao, Zhexiong Wan, Yuchao Dai, *Member, IEEE* and Xin Yu, *Member, IEEE*

Abstract—Single image blind deblurring is highly ill-posed as neither the latent sharp image nor the blur kernel is known. Even though considerable progress has been made, several major difficulties remain for blind deblurring, including the trade-off between high-performance deblurring and real-time processing. Besides, we observe that current single image blind deblurring networks cannot further improve or stabilize the performance but significantly degrades the performance when re-deblurring is repeatedly applied. This implies the limitation of these networks in modeling an ideal deblurring process. In this work, we make two contributions to tackle the above difficulties: (1) We introduce the idempotent constraint into the deblurring framework and present a deep idempotent network to achieve improved blind non-uniform deblurring performance with stable re-deblurring. (2) We propose a simple yet efficient deblurring network with lightweight encoder-decoder units and a recurrent structure that can deblur images in a progressive residual fashion. Extensive experiments on synthetic and realistic datasets prove the superiority of our proposed framework. Remarkably, our proposed network is nearly **6.5× smaller** and **6.4× faster** than the state-of-the-art while achieving comparable high performance.

Index Terms—Idempotent Network, Single Image Blind De-blurring, Efficient Deblurring

I. INTRODUCTION

SINGLE image blind deblurring aims at estimating a sharp image from a blurry input. It is a challenging ill-posed problem as both the sharp image and the kernel need to be estimated [12]–[15]. This problem becomes even more challenging for real-world complex blurry images when different exposure times, camera motion, and multiple moving objects exist. To tackle these difficulties, traditional optimization-based approaches estimate the sharp image and the kernel alternatively, where various priors have been exploited to regularize this procedure [12]–[14], [16], [17]. Recently, remarkable progress has been made by designing different network architectures to learn the mapping from a blurred image to its corresponding sharp version in an end-to-end manner without estimating the blur kernels [1], [3], [4], [8]. These approaches have achieved profound success on benchmark datasets such as GoPro [1]. Despite the advance, a very recent study [18] reports that state-of-the-art models trained on the synthetic dataset do not generalize well to real-world blurry images. Therefore, it is still challenging for existing methods to address complex non-uniform motion blurs in dynamic scenes.

Yuxin Mao, Zhexiong Wan, and Yuchao Dai are with School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shaanxi, 710129, China. Yuxin Mao and Zhexiong Wan contributed equally. Yuchao Dai (daiyuchao@gmail.com) is the corresponding author.

Xin Yu is with Faculty of Engineering and Information Technology, the University of Technology Sydney, Sydney, NSW 2007, Australia.

Digital version are available at https://doi.org/10.1109/TCSVT.2022.3202361

Fig. 1. **Speed vs Performance.** Each circle represents the performance of a model in terms of FPS and PSNR on the GoPro [1] dataset with 1280 × 720 images using an RTX 2080Ti GPU. The radius of each circle denotes the model's number of parameters. Our method achieves high performance with real-time runtime and small parameters compared with state-of-the-art blind deblurring methods including MS-CNN [1], SRN [2], DMPHN [3], DeblurGAN [4], DeblurGANv2 [5], Gao*et al.* [6], MT-RNN [7], SAPHN [8], RADN [9], MSCAN [10] and MPRNet [11].

In practical applications, it is intuitive to determine whether any obtained image is blurred or not, but it is difficult to assert whether the image has been deblurred. Therefore, for an image deblurring system, it is inevitable to repeatedly input an image that has been deblurred. We expect the performance could be improved or maintained when a deblurred image is repeatedly input to an image deblurring model. To verify this hypothesis, we repeatedly implement the state-of-the-art methods with their pre-trained models [3], [7] and report their re-deblurring performance in terms of PSNR in Fig. 2. Surprisingly, we observe a significant performance decrease after re-deblurring. This demonstrates that existing deblurring models cannot further improve their performance by repetitively applying themselves to blurry inputs.

Although we do not expect that re-deblur a deblurred image once again would lead to a sharper image, at least it should not significantly degrade the previously deblurred result. To remedy this issue, we resort to the concept of idempotence in mathematics, *i.e.*, an operator can be applied many times and still maintain the primary results. An ideal deblurring model should also own this property, because there is a theoretical upper limit (*i.e.*, output the completely sharp image) in deblurring. Therefore, the ideal situation is that the results are consistent when we implement the algorithm repeatedly, and we call it *Idempotent Property*. To achieve this goal, we introduce an idempotent constraint into the network design and

Fig. 2. **The performance curve of repeatedly re-deblurring.** We repeatedly input the deblurred image to the network by multiple times and report the deblurring results on the GoPro dataset. Our proposed deep idempotent network achieves very stable deblurring results while the performance of all other state-of-the-art methods decreases as the repeating times increase. Note that, to keep the training settings consistent with our results without idempotent constraint, we re-trained MT-RNN [7] without their multi-temporal data augmentation.

propose our *deep idempotent network* for single image blind deblurring. The idempotent constraint aims to maintain the consistency between the deblurred image and the re-deblurred image. With the introduced constraint, our network outputs stable deblurred results even after deblurring multiple times, as shown in Fig. 2.

Moreover, many state-of-the-art blind deblurring methods have large model sizes and take long inference time, as shown in Fig. 1. To satisfy the requirements of real-time (at least 30 FPS) applications, we design a novel, efficient, and lightweight single image blind deblurring network. Our idempotent network is composed of a lightweight encoder-decode module within a progressively recurrent iteration structure. Here, we can control the number of recurrent structure to balance the deblurring efficiency and performance. The whole network is trained by applying our idempotent constraint to the outputs of the deblurring and re-deblurring processes. Once our network is trained, it only runs in a single feed-forward fashion without re-deblurring in testing.

Our proposed simple yet efficient idempotent network achieves state-of-the-art deblurring performance on the Go-Pro dataset with 1280×720 images and runs in real-time. Our idempotent constraint is quite generic, and we further demonstrate its superiority on the tasks of image dehazing and deraining. Moreover, we apply the idempotent constraint to the open-source code of the existing state-of-the-art models, and further improve their results.

Our main contributions are summarized as follows:

1) We introduce an idempotent property to single image blind deblurring, and propose an idempotent constraint to improve non-uniform deblurring performance.

2) We design a simple yet efficient deblurring network that achieves real-time and high performance by progressive residual deblurring with recurrent structure.

3) Our model, while achieving comparable high performance on the GoPro benchmark, is nearly $6.5\times$ smaller and

6.4× faster than the existing state-of-the-art approach, *i.e.*, MPRNet [11].

4) Our proposed model achieves superior generalization performance on the real captured RealBlur benchmark. The proposed idempotent network architecture and constraint can be easily generalized to dehazing and deraining and improve their performance.

II. RELATED WORK

In this section, we briefly review both optimization-based and learning-based blind image deblurring methods, and idempotence in deep learning.

Optimization based image deblurring. Existing methods based on optimization mainly focus on exploiting different image priors to recover sharp images from blurry images. These valid priors can be enumerated as sparse gradients [19], l_0 norm prior [16], patch recurrence prior [12], dark channel prior [13], bright channel prior [17], latent structure prior [20], minimal pixels prior [21] and super-pixel prior [22]. Benefiting from the hand-crafted priors, optimization-based algorithms achieve competitive deblurring results for blurry images. However, many priors are only designed for specific blurry scenes and cannot generalize to cross-domain images. Besides, Srinivasan *et al.* [23] introduce a general model for light field camera motion estimation and image deblurring, and Mohan *et al.* [24] decompose this model to achieve full-resolution motion deblurring. Meanwhile, optimization-based methods are often time-consuming and need a complex parameter-tuning strategy for different datasets, which restricts their real-world applications.

Deep image deblurring. Image deblurring greatly benefits from the progress of deep learning. Deep neural networks learn the nonlinear mapping between the blurred and sharp image pairs to deal with complex motion blur. Previous deblurring methods apply convolutional neural networks (CNN) in the process of non-blind deblurring [25], [26].

Recently, researchers shift their attention to blind deblurring [1]–[3], [6]–[8], [27]–[31]. Following a coarse-to-fine scheme, MS-CNN [1] and SRN-Deblur [2] introduce multi-scale deep networks to restore sharp images in an end-to-end manner. DeblurGAN [4], [5] and DBGAN [32] regard image deblurring as an image generation problem and use a Generative Adversarial Network (GAN) [33] for deblurring. Gao *et al.* [6] add nested skip connections and a multi-scale parameter selective sharing strategy on a network. Lumentut *et al.* [34] propose a recurrent network for full-resolution light field image deblurring. Shen *et al.* [29] propose a human-aware deblurring approach to remove the blur of foreground humans. DMPHN [3] proposes a multi-patch hierarchical network to better deal with spatially non-uniform motion blur. SAPHN [8] combines the multi-patch hierarchical structure with global attention and adaptive local filters to learn the transformation of features in the deblurring process. MSCAN [10] proposes a channel-attention convolutional neural network for single image denblurring. Wang *et al.* [35] propose a recursive video deblurring network, and MACNN [36] introduces the multi-attention mechanism to video deblurring. MPRNet [11]

proposes an effective multi-stage architecture with a cross-stage feature fusion module and supervised attention module, that progressively learns restoration functions for the degraded inputs. MT-RNN [7] designs a shared weight neural network with recurrent feature maps and proposed an incremental temporal training strategy with additional temporal data augmentation. In the training process, they used synthetic images with different levels of blurriness as their supervision, thus their network learned the ability to progressively deblur. On the contrary, as shown in Table I and Fig. 9, our proposed deep idempotent network achieves better deblurring performance and still owns the ability of progressive deblurring without using such temporal data augmentation strategy.

Idempotence in deep learning. There are few works in exploring idempotence in deep learning. Zhao *et al.* [37] propose a Merge-and-Run mapping with parallel residual branches to keep the information flow linearly idempotent, which assembles the residual branches in parallel. Xing *et al.* [38] extend the Merge-and-Run block into a semantic RGB-D segmentation task to effectively fuse two modality inputs. Different from these works, we enforce our deblurring network to achieve this idempotent property, thus keeping the deblurring and re-deblurring results consistent. They try to enforce the idempotency between the convolutional layers but do not guarantee the idempotent property for the whole network outputs, which do not achieve our goal of idempotent deblurring. Another perspective to understand the idempotent property of the model is from the fixed point theory. It can be considered as finding a fixed point in the network output space that can ensure the stability of the deblurring performance. To achieve this goal, the deep equilibrium model [39] solves the fixed point directly by a few convolutional layers. Instead of solving the fixed point analytically, we introduce a novel idempotent constraint to approximate realization of the fixed point constraints and achieve stable re-deblurring.

III. DEEP IDEMPOTENT DEBLURRING FRAMEWORK

In this section, we propose a deep idempotent deblurring framework that embeds the idempotent property into a deep network for efficient single image blind deblurring. To begin with, we first explain why idempotency is needed for an image deblurring model. Then, we present the definition of the idempotent network and introduce our idempotent network architecture in detail. Finally, we describe the proposed idempotent constraint as the training loss function.

A. Why idempotent property is needed?

Images captured in the real world may have different blurring levels from the extreme case of completely sharp (without any blur) to seriously blurred. The image deblurring algorithms deployed in a real-world system should be able to handle these situations. A straightforward idea is to design a classifier to select different deblurring models based on the blurring levels. However, this classifier requires additional computational effort and will lead to worse deblurring performance when the classification results are incorrect. Therefore,

we would prefer an end-to-end model that can better handle the complex non-uniform motion blur in practice.

In addition, the output of an ideal deblurring model should be a sharp image. If we apply the same model again on the deblurred image, it should stably output the same sharp image. We believe that an ideal deblurring model should have the idempotent property. From another perspective, the re-deblurring process can be regarded as deblurring an image with a blur level of zero. Previous work [7] has shown that a fully convolutional network can deal with different levels of motion blur at different spatial locations. Based on this, we use a uniform regularization term for each pixel with non-uniform blurring levels by introducing the idempotent constraint. We believe that the idempotent property also helps improve convolution-based models' ability to handle non-uniform motion blur. In the subsequent sections, we will present our idempotent deblurring framework in detail, which makes our network output idempotent deblurred results. Experiments show that idempotent constraint enables the model to better deal with non-uniform motion blur.

B. Definition of Idempotent Network

Mathematically, an operation $f(\cdot)$ is called idempotent if and only if $f(f(\cdot)) = f(\cdot)$. This equation means a certain operation can be applied multiple times and still maintains the primary result.

In this paper, we extend this concept to deep neural networks. Given an input $\mathbf{x}$, a network Φ with parameters Θ is called an idempotent network if and only if the outputs by the repeated implementation are the same, *i.e.*,

$$\begin{aligned} \Phi(\Phi(\mathbf{x};\Theta);\Theta) &= \Phi(\mathbf{x};\Theta); \\ \Phi^k(\mathbf{x};\Theta) &= \Phi(\mathbf{x};\Theta), \end{aligned} \tag{1}$$

where k is a positive integer, indicating the number of repeating times. The above equation implies that for a deep deblurring network, given a pair of blurry and sharp images (I_B, I_S), if we feed the blurry image I_B into the network, the deblurred results $\hat{I}$ should be idempotent nevertheless how many times deblurring operations have been applied.

C. Idempotent Network Architecture

We propose a simple yet efficient single image blind deblurring network, the structure of this network is illustrated in Fig. 3. This network uses a shared weight basic lightweight encoder-decoder module with a recurrent structure to achieve iterative residual deblurring. Then we repetitively implement this network by taking the previous deblurred output as the next deblurring input, and then apply a novel idempotent constraint to these output results. Next, we will introduce each component of our idempotent network.

Basic Encoder-Decoder Deblurring Unit. To implement a simple network structure with minimal parameters, we first use a lightweight U-Net [40] like encoder-decoder module as our basic unit for iterative residual deblurring,

$$\hat{I}^i = \Phi_{\text{Basic}}(\hat{I}^{i-1}; \Theta), \tag{2}$$

Fig. 3. **The overall framework of our idempotent deblurring network and idempotent constraint.** The deblurring network takes blurry images as input and outputs deblurred images by an iterative recurrent lightweight encoder-decoder structure. In all iterations, we only use the same basic model with shared weights and connect them by residual connection. The idempotent loss makes the outputs by repeating re-deblurring consistently in the training phase.

Fig. 4. **The structure of our encoder and decoder.** The number on each block denotes parameters of the residual block, convolution and deconvolution layers. From left to right are In Channel, Out Channel, Kernel size, and Stride. And the layers in the residual block are convolution-ReLU-convolution with skip connection.

where i is the index of the iterations of the overall structure ($i = 1, 2, ..., N$), and N is the total number of iterations. $\hat{I}^i, \hat{I}^{i-1}$ are the deblurred images at the i-th and the $(i-1)$-th iteration. Specially, when i equals to 1, $\hat{I}^{i-1}$ indicates the input blurry image. Φ_{Basic} is the basic deblurring unit, which has trainable parameters Θ.

As shown in Fig. 4, there are 15 convolutional layers with 6 residual connections [41] and 6 ReLU activation functions in the encoder and decoder, respectively. Each of these residual blocks consists of two convolution layers with stride=1, a

residual connection, and a ReLU layer, which do not change the size of feature maps. In addition, two convolution layers with stride=2 are interposed between the residual blocks in the encoder to downsample the feature maps. Correspondingly, the decoder has two transpose convolutional layers in a symmetrical position for upsampling. In all convolution layers, the kernel size is 3×3. We also add a skip connection between corresponding encoder and decoder levels to fuse feature and enhance feature representation, this can facilitate network learning. Note that the residual connection uses an addition operation, while the skip connection uses concatenation.

Progressive Residual Deblurring. Iterative stacking or multi-scale structure has been widely used in existing deblurring networks [1]–[3], [7]. To progressively achieve better deblurring performance, we use the basic deblurring unit with shared parameters to estimate the residual image for iterative deblurring. As shown in Fig. 3, each output of the deblurring unit is obtained as the sum of the input image and the residual images:

$$\hat{I}^i = \hat{I}^{i-1} + \Phi_{\text{Basic}}(\hat{I}^{i-1}; \Theta). \tag{3}$$

Given a blurry image $I_B = \hat{I}^0$ as input, our model outputs the final deblurred image $\hat{I}^N$ through iterative residual deblurring. Without using additional data supervision like MT-RNN [7], our model learns to progressively deblur the input images. A visual comparison is shown in Fig. 9, which demonstrates the progressive learning ability of our residual deblurring structure.

Feature Maps Recurrence. We add a recurrent structure to the basic residual network to establish the relationship between two adjacent iterations. This mechanism recurs feature maps F_1^{i-1}, F_2^{i-1} from the two last residual blocks in decoder after upsampling layer at the $(i-1)$-th iteration. And then concatenate F_1^{i-1}, F_2^{i-1} corresponding with the feature maps in encoder at the i-th iteration before downsampling layer. (*c.f.* Fig. 4). Specifically, this structure is modeled as:

$$\hat{I}^i, F_1^i, F_2^i = \Phi_{\text{FMR}}(\hat{I}^{i-1}, F_1^{i-1}, F_2^{i-1}; \Theta), \tag{4}$$

where F_1^{i-1}, F_2^{i-1} are the feature maps from the decoder in the $(i-1)$-th iteration as the i-th iteration input, and output feature maps F_1^i, F_2^i for next iteration.

Latent Code Recurrence. The feature map recurrence we proposed above can only retain the information between two adjacent iterations. As first used in SRN-Deblur [2], Long Short-Term Memory (LSTM) [42], [43] can embed long term blur patterns across multiple deblurring iterations. The hidden state in LSTM retains the feature information from the previous iterations. To achieve similar performance with smaller parameters and cheaper computation, we use the Gated Recurrent Unit (GRU) [44] as our Latent Code Recurrence (LCR) module in our network.

Since the network is recurrent during multiple deblurring iterations, the feature maps from the last convolution layer in the encoder of each iteration are fed into GRU as the latent code. After passing through the memory unit, the feature maps can conditionally retain and forget the information of previous iterations, then can be used as the input of the decoder to restore the residual images of this iteration. The calculation process is modeled as follows:

$$
\begin{aligned}
z^i &= \text{sigmoid}(\text{Conv}([h^{i-1}, e^i], \Theta_z)), \\
r^i &= \text{sigmoid}(\text{Conv}([h^{i-1}, e^i], \Theta_r)), \\
\hat{h}^i &= \tanh(\text{Conv}([r^i \odot h^{i-1}, e^i], \Theta_h)), \\
h^i &= (1 - z^i) \odot h^{i-1} + z^i \odot \hat{h}^i,
\end{aligned}
\tag{5}
$$

where $[\cdot]$ is the concatenation operation and $\odot$ represents the Hadamard product. z^i, r^i and $\hat{h}_i$ are the update gate, reset gate and candidate activation vector in GRU. $\Theta_z, \Theta_r, \Theta_h$ are the parameters of each convolution layer, respectively.

Thus, our basic unit for progressive residual deblurring with feature maps recurrence and latent code recurrence can be modeled as follows:

$$
\hat{I}^i, F_1^i, F_2^i, h^i = \Phi_{\text{LCR}}(\hat{I}^{i-1}, F_1^{i-1}, F_2^{i-1}, h^{i-1}; \Theta), \tag{6}
$$

where h^{i-1}, h^i represent the hidden state output by previous iteration $(i-1)$ and current iteration i.

Idempotent Re-Deblurring by Multiple Times. It is difficult for a deep network to output the same results as the input, so we propose a idempotent constraint to train our deblurring network. As shown in Fig. 3, our model first outputs the deblurring result $\hat{I}_1 = \hat{I}_1^N$, then takes it as next input and get the re-deblurring result $\hat{I}_2 = \hat{I}_2^N$ (N denotes the number of iterations) for multiple times.

Note above re-deblurring only exists in training, and in the inference phase, we only need to apply our model once to get the deblurred result. Because of the residual deblurring structure, the idempotent constraint can be satisfied when the sum of all the residual outputs is close to zero.

D. Idempotent Constraint

To keep the deblurring network output idempotent and enhance the deblurring performance with idempotent property, we enforce the idempotent constraint as a loss function to supervise the training process. We term such loss function as an idempotent loss. As shown in Fig. 3, the network is repeated

twice and we directly constrain these two deblurring outputs to be consistent. Therefore, the idempotent loss is defined by the L_1 distance as:

$$
\mathcal{L}_{Idem} = \|\hat{I}_1 - \hat{I}_2\|_1, \tag{7}
$$

where $\hat{I}_1$ and $\hat{I}_2$ denote two latent sharp image outputs by deblurring and re-deblurring.

We use pairs of blurry and sharp images (I_B, I_S) to supervise the training of our deblurring network, and then calculate the L_1 distance as our deblurring loss at each output by re-deblurring:

$$
\mathcal{L}_{Sharp} = \sum_{j=1}^{2} \alpha_j \|\hat{I}_j - I_S\|_1. \tag{8}
$$

The final objective function is reached as:

$$
\mathcal{L} = \lambda \mathcal{L}_{Idem} + \mathcal{L}_{Sharp}, \tag{9}
$$

where α_j and λ are the trade-off parameters.

IV. EXPERIMENTS

A. Experimental Details

Datasets. Following the general setting of single image deblurring task [1]–[4], [7], [9], [11], [30], we use the GoPro dataset [1] to train our proposed model. The blurry images of the GoPro dataset are synthesized by averaging different numbers (7–13) of successive latent frames from 240 FPS video sequences captured by a GoPro Hero 4 camera. As a common benchmark for image motion deblurring, it contains 3,214 blurry-sharp image pairs. We follow the widely used split method as [1] and use 2,103 pairs from the linear subset for training and the remaining 1,111 pairs as the test set for evaluation.

We also evaluate the generalization ability of our method on a real-world blurry scenes dataset, *i.e.*, RealBlur [18], a commonly used dataset in recent years. It contains two versions named as *RealBlur-J* (JPEG compressed) and *RealBlur-R* (RAW), where each version includes 980 pairs of geometrically aligned real-world blurry and ground-truth sharp images captured by a well-designed image acquisition system composed of a beam splitter. Following the original settings of the RealBlur dataset, we also conduct photometric alignment between the outputted deblurred images and ground-truth sharp images before computing PSNR and SSIM.

Implementation Details. We implement our model[1] by PyTorch [45] with two NVIDIA RTX 2080Ti GPU for training and one for evaluation. The trade-off parameter α_1, α_2 is set to 1.0 and λ is 0.1, respectively. We define the number of iterations N as 6. All weights are initialized from scratch by the Xavier method [46]. The Adam [47] solver is used to optimize our network for 3,000 epochs with default parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The training mini-batch size is set to 6. The initial learning rate is 10^{-4} and decays by 0.5 after every 500 epochs. Following DMPHN [3], the input blurry images are normalized to range $[0, 1]$ and subtracted by 0.5. In training, the input blurry images and corresponding

---

[1]Our code and pre-trained model will be released.

TABLE I
QUANTITATIVE RESULTS ON THE GOPRO TEST DATASET [1] IN TERMS OF PSNR, SSIM, NUMBER OF PARAMETERS AND INFERENCE TIME. THE 1ST, 2ND AND 3RD BEST PERFORMANCES ARE HIGHLIGHTED WITH RED, BLUE AND GREEN (BEST VIEWED IN COLOR).

Methods	PSNR (dB)	SSIM	Param.	Time(s)
Sun *et al.* [27]	24.64	0.843	-	-
MS-CNN [1]	29.23	0.914	21.0M	0.94
DeblurGAN [4]	28.70	0.958	3.50M	0.12
DeblurGANv2 [5]	29.55	0.934	60.9M	0.058
SRN-Deblur [2]	30.26	0.931	3.76M	0.25
Gao *et al.* [6]	30.92	0.942	2.84M	0.76
MT-RNN [7]	31.15	0.945	2.63M	0.048
DMPHN [3]	30.25	0.935	7.23M	0.032
Stack(4)-DMPHN [3]	31.20	0.945	28.9M	0.35
Stack(2)-VMPHN [3]	31.50	0.948	28.9M	(0.55)
MSCAN [10]	31.24	0.945	7.5M	0.42
RADN [9]	31.76	0.953	-	(0.038)
SAPHN [8]	31.85	0.948	24.0M	(0.28)
MPRNet(1) [11]	30.43	-	5.6M	(0.04)
MPRNet(2) [11]	31.81	-	11.3M	(0.08)
MPRNet(3) [11]	32.66	0.959	20.1M	0.15
Ours(w/o Idem.)	31.80	0.949	3.11M	0.028
Ours(w/ Idem.)	31.92	0.953	3.11M	0.028

TABLE II
QUANTITATIVE ANALYSIS ON THE *RealBlur* TEST DATASET [18] FOR MODELS ONLY PRE-TRAINED ON GOPRO DATASET [1]

Methods	RealBlur-J		RealBlur-R		
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	Time(s)
MS-CNN [1]	27.87	0.827	32.51	0.841	0.77
Stack(4)-DMPHN [3]	27.80	0.847	35.48	0.947	0.29
DeblurGAN [4]	27.97	0.834	33.79	0.903	0.098
MT-RNN [7]	28.44	0.862	35.77	0.951	0.039
SRN-Deblur [2]	28.56	0.867	35.66	0.947	0.20
DeblurGANv2 [5]	28.70	0.867	35.26	0.944	0.048
MPRNet(3) [11]	28.70	0.873	**35.99**	**0.952**	0.16
Ours(w/o Idem.)	28.69	0.868	35.79	0.949	0.023
Ours(w/ Idem.)	**28.72**	**0.876**	35.86	0.951	**0.023**

TABLE III
QUANTITATIVE ANALYSIS ON MULTI-BLURRING LEVEL SYNTHETIC DATASET. DIFFERENT NUMBER IN THE FIRST ROW MEANS THE NUMBER OF SUCCESSIVE LATENT FRAMES USED TO SYNTHESIZE THE BLURRY IMAGES.

Methods \ PSNR(dB)	5	7	9	11	13	15
MT-RNN [7]	**35.17**	33.67	32.37	31.12	29.91	28.68
Stack(4)-DMPHN [3]	33.15	32.90	32.26	31.19	29.85	27.73
Ours(w/o Idem.)	34.77	33.82	32.81	31.75	30.55	29.14
Ours(w/ Idem.)	35.05	**33.98**	**33.00**	**31.92**	**30.72**	**29.31**

ground-truth sharp images are randomly cropped to 256×256 pixels patches. Additionally, we randomly rotate and/or flip the image patches for data augmentation. The color saturation is also randomly changed on the input images for robust learning.

B. Comparison with the State-of-the-art Methods

In this subsection, we perform quantitative and qualitative studies with existing state-of-the-art methods on the benchmark datasets (*i.e.* GoPro and RealBlur).

Quantitative Evaluations. Following the common experimental setting as previous works [3], [7]–[9], we compare the performance and generalization ability across different datasets of our deep idempotent deblurring network with previous state-of-the-art deblurring methods in a quantitative way. The experimental results in terms of *PSNR*, *SSIM*, *Parameters* and *Time* for different deblurring methods on GoPro test dataset are shown in Table I. For a fair comparison, we perform the experiments of running time on a single RTX 2080Ti GPU. Except for Stack(2)-VMPHN [3], RADN [9] and MPRNet-(1)&(2) [11] are from their paper which do not provide the opensource code or the corresponding model, SAPHN [8] are provided by the authors (bracketed in Table I).

On the GoPro test dataset, our method achieves comparable high performance to state-of-the-art methods with smaller parameters and faster inference time. In particular, although MPRNet(3) [11] achieves higher performance, our model is nearly $6.5\times$ smaller and $6.4\times$ faster than it. The comparison with MPRNet(2) shows that we achieve equivalent performance with fewer parameters and shorter inference time. Moreover, the adversarial loss improves the visual quality but may sacrifice the pixel-wise metric results, while the SSIM metric prefers the structural similarity rather than the pixel-wise intensity similarity. Therefore, the GAN-based methods [4] tend to achieve better SSIM than other methods trained with the L_1 loss. And the L_1 loss often leads to smooth results.

We also perform a quantitative comparison of generalization results on the RealBlur [18] dataset for models only pre-trained on the GoPro dataset. The quantitative results are reported in Table II. We can observe that our model performs better compared to previous state-of-the-art methods, which confirms that our model is more robust in real-world scenes with better generalization ability across different datasets for image deblurring. Note that these results are from [18], except for MT-RNN [7] which is reproduced by us. Due to the different image sizes of the RealBlur dataset, we test the average inference time of each model, and our model maintains a very fast inference speed.

To further demonstrate the deblurring performance and generalization ability for different blur levels, we resynthesized a multi-blurring level dataset following the synthesis pipeline of the GoPro dataset [1]. Quantitative results in Table III show that our model achieves better deblurring performance on the multi-blurring level dataset, indicating the superiority of our idempotent framework for dynamic scene non-uniform deblurring. Note that MT-RNN [7] uses blurry images of different blur levels for supervision during the training process. This strategy makes MT-RNN more advantageous when using datasets synthesized at blur level 5 or 7, and causes the performance of our network to be inferior to MT-RNN when the blur level is 5. However, when the blur level is 7-15, the performance of our proposed idempotent deblurring network is significantly better than MT-RNN and DMPHN.

Qualitative Evaluations. We perform the visual quality comparison of deblurred images by our proposed model and recent CNN-based dynamic scene deblurring networks, including MT-RNN [7] and Stack(4)-DMPHN [3] (Considering the space, we only tested the best of the two open-source methods). Fig. 5 shows several blurry images from the GoPro test

(a) Blur input   (b) MT-RNN [7]   (c) Stack(4)-DMPHN [3]   (d) Ours

Fig. 5. **Visual comparisons on the GoPro testing dataset.** Column (a) is the original blurry images, (b), (c), (d) are the deblurring results from MT-RNN [7], Stack(4)-DMPHN [3] and our method, respectively. Best Viewed on Screen.

(a) Blur input   (b) MT-RNN [7]   (c) Stack(4)-DMPHN [3]   (d) Ours

Fig. 6. **Visual comparisons on the RealBlur dataset.** Column (a) is the original blurry images, (b), (c), (d) are the deblurring results from MT-RNN [7], Stack(4)-DMPHN [3] and our method, respectively. Best Viewed on Screen.

dataset and their corresponding deblurring results produced by the above methods. We can observe that although the above methods can play a good deblurring effect, the handling of some details such as blurred structure recovery and blurred edges is not satisfactory. For example, on the first row in Fig. 5, our proposed model can better handle highly blurred scenes,

(a) Blur input

(b) MT-RNN [7]

(c) Stack(4)-DMPHN [3]

(d) Ours

Fig. 7. **Visual comparisons on the motion blurred thermal samples from** [48]. Column (a) is the original blurred thermal images, (b), (c), (d) are the deblurring results from MT-RNN [7], Stack(4)-DMPHN [3] and our method, respectively. Best Viewed on Screen.

especially in the zoom-in region (Such as recovering the structure of the "window" is better than Stack(4)-DMPHN). And on the second row in Fig. 5, our proposed model can also perform better on recovering the blurred edges caused by the large depth of field and highly dynamic moving objects.

We compare the qualitative results on the RealBlur test datasets, as shown in Fig. 6. We can observe that our proposed model has a very outstanding advantage for deblurring the text in the scene accompanied by uneven lighting, while there are still noticeable artifacts for the results of MT-RNN and Stack(4)-DMPHN. Moreover, our model performs better for deblurring faces (Fig. 6 second row) and tiny objects with intricate details (Fig. 6 third row).

Our proposed idempotent deblurring network can be applied to other imaging modalities. Following [48], we perform a qualitative analysis of our pre-trained model and MT-RNN, Stack(4)-DMPHN on motion blurred thermal images on the dataset of [48]. The visualization comparisons are shown in Fig. 7. We can observe that our model recovers sharper details from the blurred thermal inputs, especially for highly dynamic moving objects and structural details of the image. The experiments on the thermal images show that our proposed deblurring framework can generalize well to other image modalities.

C. Ablation studies

In this section, we perform ablation studies on the GoPro test dataset to analyze the effectiveness of each component of our proposed method. The relevant experimental results are reported in Table IV. *FMR* represents whether using feature map recurrence. *LCR* denotes whether to use latent code recurrence to embed states during iterations. *Times* means the deblurring times in training, including deblurring and re-deblurring. *Idem.* means whether using an idempotent constraint on the repeating

TABLE IV
QUANTITATIVE ANALYSIS OF ABLATION STUDIES.

	FMR	LCR	Times	Idem.	Iters	PSNR (dB)	SSIM
(a)	-	-	1	-	1	29.463	0.9259
(b)	-	-	1	-	2	29.958	0.9323
(c)	-	-	1	-	4	30.933	0.9434
(d)	-	-	1	-	6	31.337	0.9461
(e)	-	-	1	-	8	31.464	0.9468
(f)	✓	-	1	-	6	31.479	0.9468
(g)	✓	-	2	-	6	31.403	0.9465
(h)	✓	-	2	✓	6	31.521	0.9469
(i)	✓	✓	1	-	6	31.796	0.9487
(j)	✓	✓	2	-	6	31.684	0.9471
(k)	✓	✓	3	✓	6	31.892	0.9523
(l)	✓	✓	2	✓	6	31.917	0.9527
(m)	✓	✓	2	✓	8	31.972	0.9529

re-deblurring outputs. *Iters* is the total number of iterations N in a single deblurring process.

Effectiveness of the Idempotent Constraint. We first conduct two experiments to evaluate the effectiveness of the idempotent constraint. As shown in Table IV (*i*) and (*l*), after using the idempotent constraint during the training process, the deblurring performance can be improved from 31.796dB to 31.917dB in terms of PSNR. Real-world deblurring performance on the RealBlur dataset reported in Table II also demonstrates the effectiveness of our idempotent constraint. If we only perform re-deblur and use sharp loss $\mathcal{L}_{Sharp}$ to supervise without idempotent loss $\mathcal{L}_{Idem}$ (*c.f.* Table IV (*j*)), the result will be severely reduced. We analyze that directly repeating the whole deblurring model will cause a bottleneck in the information flow, which explains the inferior result of (*j*) to (*i*), and demonstrates the effectiveness of our proposed idempotent constraint. We experiment with deblurring 3 times with the idempotent constraint. The results of (*k*) and (*j*)

| (a) MT-RNN [7] | (b) Stack(4)-DMPHN [3] | (c) Ours | (d) Ground-Truth |

Fig. 8. **Visual comparisons of model idempotence**. Columns 1 to 3 show the deblurred results on the GoPro dataset when re-deblurring 10 times, and column 4 is the ground-truth sharp image for visual comparison.

show a significant performance improvement. This comparison demonstrates the effectiveness of the idempotent constraint in deblurring multiple times. However, as the performance of (k) is comparable to (l), considering the faster training speed, we set the deblurring times of our idempotent framework to 2.

In Fig. 2, compared with others, our model trained with the idempotent constraint could retain stable performance when re-deblurring multiple times. We compare our model with MT-RNN [7] and Stack(4)-DMPHN [3] by repeating the re-deblurring process for 10 times. As shown in Fig. 8, after repeating 10 times, more noise appears in the outputs of Stack(4)-DMPHN while the result of MT-RNN shows overly smoothed and unrealistic effects. In contrast, our model with the idempotent constraint produces a visually realistic result that is close to the sharp ground-truth image. The re-deblurring results are repeated 10 times just to highlight the degradation trend of these methods. We only need to repeat twice in training, and for real application, the deblurring algorithm only needs to be done once *i.e.* no re-deblurring. Benefiting from the idempotent constraint, our model maintains stable idempotence for multiple re-deblurring results.

Effectiveness of Progressive Residual Deblurring. By training the model with progressive residual deblurring, we allow the network to consider wider image contexts and gradually restore the sharp image. In Table IV, we experiment on the influence of iterations when training the progressively deblurring model. As shown in Line (a)-(e), our model achieves improved performance as the iteration number increase, which validates the effectiveness of the progressive residual learning mechanism. The results in Line (k) and (l) for iterations 6 and 8 also demonstrate this. The iteration number is a hyperparameter that we can manually choose to achieve either better deblurring performance or faster inference. Considering the trade-off between inference time and deblurring performance, we set the iteration number to 6.

Effectiveness of the Feature Maps Recurrence. We observe that results become better with the help of the feature maps recurrence structure in Table IV (d) and (f). Because it inherently passes the high-frequency features from the decoder to the next encoder, the high-frequency features will be further emphasized. The highlighted high-frequency features thus lead to better deblurring performance.

Effectiveness of the Latent Code Recurrence. In Table IV (h) and (k), the latent code recurrence improves PSNR by

TABLE V
QUANTITATIVE RESULTS OF RETRAINED **DMPHN** AND **MT-RNN**
WITH OUR IDEMPOTENT CONSTRAINT.

Methods	DMPHN	DMPHN(w/ Idem)	MT-RNN	MT-RNN(w/ Idem)
PSNR (dB)	30.25	30.36	31.15	31.28
SSIM	0.935	0.936	0.945	0.945

TABLE VI
QUANTITATIVE ANALYSIS OF RE-DEBLURRING PERFORMANCE. THE
FIRST TWO COLUMNS REPRESENT USING DIFFERENT METHODS TO
DEBLUR THE ORIGINAL BLURRY IMAGES FROM THE GOPRO DATASET.
THE REMAINING THREE COLUMNS REPRESENT EACH METHOD'S
DEBLURRING RESULTS WHEN RE-DEBLURRING THE DEBLURRED IMAGES
BY THE FIRST COLUMN'S METHODS.

Deblur Method	Deblur	Re-deblur Method	1st Re-deblur	2nd Re-deblur
Ours	31.917	Ours	31.916	31.914
Stack(4)-DMPHN [3]	31.402	Stack(4)-DMPHN [3]	30.192	29.896
		Ours	31.370	31.369
MTRNN [7]	31.149	MTRNN [7]	30.538	30.322
		Ours	31.041	31.039

about 0.396dB. This demonstrates that our LCR indeed mitigates the problem of fewer channel numbers at the bottleneck caused by the lightweight network. The GRU module encodes blurry patterns into hidden states among multiple iterations, thus the progressive residual deblurring process can fully utilize features that are more helpful to restore the sharp image.

D. Idempotent Property on Previous Methods

To demonstrate the effectiveness of our proposed idempotent constraint, we retrain DMPHN and MT-RNN with our idempotent constraint. The evaluation results on the GoPro test dataset are reported in Table V. Our proposed idempotent constraint can improve the deblurring performance of these two methods without any modification of the network.

To verify the robustness of our method when dealing with deblurred images, we further perform experiments by feeding images obtained from other deblurring methods into our model. In Table VI, we report the re-deblurring performance of our model and the re-deblurring performance of Stack(4)-DMPHN and MT-RNN. We can observe that the re-deblurring performance of our method is significantly better than Stack(4)-DMPHN and MT-RNN. This demonstrates that our proposed idempotent framework can effectively deal with

Fig. 9. **Progressively deblurred results of our method and MT-RNN [7].** Same as MTRNN, our model can achieve progressive iterative deblurring over multiple iterations, but without their temporal data augmentation in the training process.

TABLE VII
QUANTITATIVE ANALYSIS OF ITERATIVE RESIDUAL DEBLURRING COMPARED WITH **MT-RNN [7]**.

Method	Deblurring Times	Calculation Data	1	2	3	4	5	6
Ours	1	PSNR (dB)	25.475	25.556	26.347	28.191	30.975	31.917
		Every	3.614	4.992	3.834	3.123	4.302	3.903
		Sum	3.614	6.235	5.492	8.329	11.674	13.026
	2	PSNR (dB)	31.903	31.903	31.897	31.895	31.895	31.973
		Every	0.359	0.422	0.142	0.381	0.675	0.673
		Sum	0.359	0.298	0.540	0.531	0.359	0.071
MT-RNN [7]	1	PSNR (dB)	26.432	27.606	29.001	30.363	31.083	31.127
		Every	2.324	3.118	3.283	3.203	2.572	1.217
		Sum	2.324	5.102	7.811	10.226	11.943	12.469
	2	PSNR (dB)	31.093	31.002	30.884	30.762	30.637	30.509
		Every	0.386	0.535	0.584	0.586	0.580	0.579
		Sum	0.386	0.803	1.175	1.491	1.765	2.020

images acquired from arbitrary sources (*i.e.*, original blurry images and deblurred images from other deblurring methods).

E. Comparison of Our Method and MT-RNN

Fig. 9 shows the output of progressively deblurred results of our method and MT-RNN [7] in different deblurring iterations. Our model archives progressively deblurring similar to MT-RNN, but do not need their temporal data augmentation. The degradation curve of MT-RNN in Fig. 2 is relatively more stable than other methods without our idempotent constraint (except for Ours (w/Idem.)). We believe this is mainly because they train the model using additional temporally augmented data. This data is composed of 3 to 13 (odd numbers) frames respectively, which is more than the commonly used split by GoPro [1]. The results of our re-trained MT-RNN without their data augmentation also prove this belief.

We report the quantitative results of iterative residual deblurring from every deblurring unit on the GoPro test dataset [1] in Table VII. The first deblurring (Deblurring Times 1) uses the blurry image as the model input, and the second deblurring (Deblurring Times 2) uses the deblurred image at the first time as the input. *PSNR* represents the distortion between the deblurred image at each iteration and the ground-truth sharp image. *Every* is the absolute value of the residual image output by the network in each iteration. *Sum* is the absolute value of the sum of the residual images from the current and previous iterations, which is equivalent to the difference between the current deblurred image and the input image of the network at the first iteration. Note that these absolute values are defined in the image value range from [0, 255].

For the first deblurring time, the deblurring performance improves with the increase of the number of iterations. Although the trend is similar, the variation (Every and Sum) of each iteration is quite different. In the process of the second deblurring (*i.e.*, re-deblurring), our performance is more stable and the PSNR of the final output is better.

In particular, the value of each residual is very small, and the sum of the residual is very close to zero. As mentioned in Section III-C, the idempotent constraint can be satisfied when the sum of all the residual outputs is close to zero in the re-deblurring process. On the contrary, the Sum value of MT-RNN [7] increases with multiple iterations step by step. This also leads to its worse and unstable performance in the second deblurring process and shows the effectiveness of our *idempotent constraint* by comparison.

(a) Hazy input　　(b) Grid-DehazeNet [49]　　(c) Ours　　(d) Ground-Truth

Fig. 10. **Visual comparisons on the SOTS indoor dataset for image dehazing.** Column (a) is the input hazy images. (b), (c) is from Grid-DehazeNet [49] and our method, respectively. (d) is the ground-truth image. Best Viewed on Screen.

(a) Rainy input　　(b) PreNet [50]　　(c) Ours　　(d) Ground-Truth

Fig. 11. **Visual comparisons on the R100H dataset for image deraining.** Column (a) is the input rainy images. (b), (c) is from PreNet [50] and our method, respectively. (d) is the ground-truth image. Best Viewed on Screen.

TABLE VIII
QUANTITATIVE ANALYSIS ON THE SYNTHESIZED GOPRO DATASET WITH DIFFERENT LEVELS OF GAUSSIAN NOISE. THE DIFFERENT NUMBERS IN THE FIRST ROW MEAN THE MAGNITUDE OF THE GAUSSIAN NOISE ADDED ON THE INPUT BLURRY IMAGES.

Methods \ PSNR(dB)	Original	$\sigma = 5$	$\sigma = 10$	$\sigma = 15$	$\sigma = 20$
Ours(w/o Idem.)	31.80	30.76	29.58	28.16	26.05
Ours(w/ Idem.)	31.92	30.89	29.66	28.24	26.55

F. Noise Adaptation of Our Method

To verify the robustness of our model to noise, we analyze the performance of our pre-trained model by adding different levels of Gaussian noise to the blurry input of the GoPro dataset [1]. The noise levels include 5, 10, 15 and 20, where each number represents the standard deviation of the normal distribution noise within the pixel range of [0, 255]. Then we evaluate the deblurring performance of our pre-trained model with or without our proposed idempotent constraint to analyze the noise adaptability. The deblurring performance under different noise levels is reported in Table VIII. Experimental results show the stability of our model for noisy inputs. This observation illustrates that our proposed idempotent constraint can make the model more robust to noise than the model without the constraint.

G. Extension to Image Dehazing

Our proposed deep idempotent framework is rather general and not limited to image deblurring. We perform image dehazing with our proposed model and the idempotent constraint to investigate the versatility and scalability of different image restoration tasks. Following the same training pipeline of Grid-DehazeNet [49], our model is trained on Indoor Training Set (ITS) and tested on the Synthetic Objective Testing Set (SOTS) Indoor Subset in RESIDE dataset [51]. Table IX shows that with the idempotent constraint, our model achieves state-of-the-art performance on the SOTS indoor dataset. These results show that our idempotent constraint can boost the dehazing performance by a large margin (1.86dB). Fig. 11 shows the visual comparison of dehazing results on SOTS indoor dataset. Compared with Grid-DehazeNet [49], our model has a clear advantage in visual effects. For instance, previous methods may not fully dehaze some image regions and produce black artifacts in the zoom-in areas.

TABLE IX
QUANTITATIVE RESULTS OF APPLYING OUR IDEMPOTENT FRAMEWORK ON THE SOTS INDOOR DATASET FOR IMAGE DEHAZING. BEST AND SECOND-BEST SCORES ARE HIGHLIGHTED AND UNDERLINED.

Methods	[52]	[53]	[49]	[54]	[55]	**Ours**(w/o)	**Ours**(w/)
PSNR (dB)	19.82	30.23	32.16	36.39	36.56	34.78	**36.64**
SSIM	.8209	.9800	.9836	.9556	**.9905**	.9823	.9851

H. Extension to Image Deraining

We also extend our proposed deep idempotent framework to image deraining. Following the same training pipeline of

TABLE X
QUANTITATIVE RESULTS OF APPLYING OUR IDEMPOTENT FRAMEWORK FOR IMAGE DERAINING. BEST AND SECOND-BEST SCORES ARE HIGHLIGHTED AND UNDERLINED.

Methods	Test100 [56]		Rain100H [57]		Rain100L [57]	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
DerainNet [58]	22.77	0.810	14.92	0.592	27.03	0.884
SEMI [59]	22.35	0.788	16.56	0.486	25.03	0.842
DIDMDN [60]	22.56	0.818	17.35	0.524	25.23	0.741
UMRL [61]	24.41	0.829	26.01	0.832	29.18	0.923
RESCAN [62]	25.00	0.835	26.36	0.786	29.80	0.881
PreNet [50]	24.81	0.851	26.77	0.858	32.44	0.950
MSPFN [63]	27.50	0.876	28.66	0.860	32.40	0.933
Ours(w/o Idem.)	28.94	0.889	29.97	0.881	34.14	0.942
Ours(w/ Idem.)	**29.00**	**0.892**	**30.10**	**0.882**	**34.68**	**0.954**

MSPFN [63], we train our model on about 13,700 clean/rain image pairs collected from [56], [57]. We evaluate the performance on the testing datasets, including Test100 [56], Rain100H [57] and Rain100L [57]. The results in Table X show that our designed structure and idempotent constraint boost the deraining performance, respectively. Fig. 10 shows the visual comparison of deraining results on R100H dataset. Compared with PreNet [50], our model has better visual effects.

V. CONCLUSION

In this paper, we have presented a novel deep idempotent network for efficient single image blind deblurring. First, we introduced the idempotent constraint to the deep deblurring network, which improves the non-uniform deblurring performance and achieves stable results w.r.t. re-deblurring multiple times. Second, we designed a simple yet efficient deblurring network through progressive residual deblurring with recurrent structure. Our model achieves state-of-the-art performance with smaller parameters and faster inference time than the state-of-the-art methods. The introduced idempotent constraint, as a regularization term, plays an important role in reducing the solution search space and thus leads to a more stable solution regardless of the times of re-deblurring. By adopting our idempotent constraint in model training, our model shows great generalization performance on real-world and synthetic datasets. Our framework is not limited to image deblurring, and we have verified the superiority of our framework in image dehazing and image deraining. In the future, we will further extend it to other image restoration tasks such as image denoising [64]. It is also promising to explore a tailored multiple-image input idempotent network for video [35] or light field image deblurring [23].

REFERENCES

[1] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3883–3891. 1, 2, 4, 5, 6, 10, 12

[2] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8174–8182. 1, 2, 4, 5, 6

[3] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5978–5986. 1, 2, 4, 5, 6, 7, 8, 9

[4] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8183–8192. 1, 2, 5, 6

[5] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 8878–8887. 1, 2, 6

[6] H. Gao, X. Tao, X. Shen, and J. Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3848–3856. 1, 2, 6

[7] D. Park, D. U. Kang, J. Kim, and S. Y. Chun, "Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

[8] M. Suin, K. Purohit, and A. Rajagopalan, "Spatially-attentive patch-hierarchical network for adaptive motion deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3606–3615. 1, 2, 6

[9] K. Purohit and A. Rajagopalan, "Region-adaptive dense network for efficient motion deblurring," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 11 882–11 889. 1, 5, 6

[10] S. Wan, S. Tang, X. Xie, J. Gu, R. Huang, B. Ma, and L. Luo, "Deep convolutional-neural-network-based channel attention for single image dynamic scene blind deblurring," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 31, no. 8, pp. 2994–3009, 2021. 1, 2, 6

[11] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 5, 6

[12] T. Michaeli and M. Irani, "Blind deblurring using internal patch recurrence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 783–798. 1, 2

[13] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, "Blind image deblurring using dark channel prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1628–1636. 1, 2

[14] L. Pan, R. Hartley, M. Liu, and Y. Dai, "Phase-only image based kernel estimation for single image blind deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6034–6043. 1

[15] A. Kaufman and R. Fattal, "Deblurring using analysis-synthesis networks pair," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5811–5820. 1

[16] L. Xu, S. Zheng, and J. Jia, "Unnatural L_0 sparse representation for natural image deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1107–1114. 1, 2

[17] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao, "Image deblurring via extreme channels prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4003–4011. 1, 2

[18] J. Rim, H. Lee, J. Wona, and S. Cho, "Real-world blur dataset for learning and benchmarking deblurring algorithms," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 184–201. 1, 5, 6

[19] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, 2006. 2

[20] Y. Bai, H. Jia, M. Jiang, X. Liu, X. Xie, and W. Gao, "Single-image blind deblurring using multi-scale latent structure prior," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 30, no. 7, pp. 2033–2045, 2020. 2

[21] F. Wen, R. Ying, Y. Liu, P. Liu, and T.-K. Truong, "A simple local minimal intensity prior and an improved algorithm for blind image deblurring," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 31, no. 8, pp. 2923–2937, 2021. 2

[22] B. Luo, Z. Cheng, L. Xu, G. Zhang, and H. Li, "Blind image deblurring via superpixel segmentation prior," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 32, no. 3, pp. 1467–1482, 2022. 2

[23] P. P. Srinivasan, R. Ng, and R. Ramamoorthi, "Light field blind motion deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3958–3966. 2, 12

[24] M. Mohan and A. Rajagopalan, "Divide and conquer for full-resolution light field deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6421–6429. 2

[25] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 1790–1798. 2

[26] J. Zhang, J. Pan, W.-S. Lai, R. W. Lau, and M.-H. Yang, "Learning fully convolutional networks for iterative non-blind deconvolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3817–3825. 2

[27] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 769–777. 2, 6

[28] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, "Neural blind deconvolution using deep priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3341–3350. 2

[29] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao, "Human-aware motion deblurring," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 5572–5581. 2

[30] J. Cai, W. Zuo, and L. Zhang, "Dark and bright channel prior embedded network for dynamic scene deblurring," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 6885–6897, 2020. 2, 5

[31] Y. Yuan, W. Su, and D. Ma, "Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3555–3564. 2

[32] K. Zhang, W. Luo, Y. Zhong, L. Ma, B. Stenger, W. Liu, and H. Li, "Deblurring by realistic blurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2737–2746. 2

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2672–2680. 2

[34] J. S. Lumentut, T. H. Kim, R. Ramamoorthi, and I. K. Park, "Deep recurrent network for fast and full-resolution light field deblurring," *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1788–1792, 2019. 2

[35] X. Zhang, R. Jiang, T. Wang, and J. Wang, "Recursive neural network for video deblurring," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 31, no. 8, pp. 3025–3036, 2021. 2, 12

[36] X. Zhang, T. Wang, R. Jiang, L. Zhao, and Y. Xu, "Multi-attention convolutional neural network for video deblurring," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 32, no. 4, pp. 1986–1997, 2022. 2

[37] L. Zhao, M. Li, D. Meng, X. Li, Z. Zhang, Y. Zhuang, Z. Tu, and J. Wang, "Deep convolutional neural networks with merge-and-run mappings," in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, 2018. 3

[38] Y. Xing, J. Wang, X. Chen, and G. Zeng, "Coupling two-stream rgb-d semantic segmentation network by idempotent mappings," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 1850–1854. 3

[39] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019, p. 688–699. 3

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on*

Medical Image Computing and Computer-Assisted Intervention (MIC-CAI), 2015, pp. 234–241. 3

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. 4

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 5

[43] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 802–810. 5

[44] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734. 5

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 8026–8037. 5

[46] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 249–256. 5

[47] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015. 5

[48] M. S. Ramanagopal, Z. Zhang, R. Vasudevan, and M. J. Roberson, "Pixel-Wise Motion Deblurring of Thermal Videos," in *Proceedings of Robotics: Science and Systems*, July 2020. 8

[49] X. Liu, Y. Ma, Z. Shi, and J. Chen, "Griddehazenet: Attention-based multi-scale network for image dehazing," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 7314–7323. 11, 12

[50] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3937–3946. 11, 12

[51] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, "Benchmarking single-image dehazing and beyond," *IEEE Transactions on Image Processing (TIP)*, vol. 28, no. 1, pp. 492–505, 2019. 12

[52] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "Dehazenet: An end-to-end system for single image haze removal," *IEEE Transactions on Image Processing (TIP)*, vol. 25, no. 11, pp. 5187–5198, 2016. 12

[53] D. Chen, M. He, Q. Fan, J. Liao, L. Zhang, D. Hou, L. Yuan, and G. Hua, "Gated context aggregation network for image dehazing and deraining," in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1375–1383. 12

[54] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia, "Ffa-net: Feature fusion attention network for single image dehazing," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 11 908–11 915. 12

[55] Q. Deng, Z. Huang, C.-C. Tsai, and C.-W. Lin, "Hardgan: A haze-aware representation distillation gan for single image dehazing," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 722–738. 12

[56] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 30, no. 11, pp. 3943–3956, 2020. 12

[57] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3855–3863. 12

[58] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing (TIP)*, vol. 26, no. 6, pp. 2944–2956, 2017. 12

[59] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3877–3886. 12

[60] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 695–704. 12

[61] R. Yasarla and V. M. Patel, "Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8405–8414. 12

[62] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 254–269. 12

[63] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8346–8355. 12

[64] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing (TIP)*, vol. 26, no. 7, pp. 3142–3155, 2017. 12

Yuxin Mao is currently a PhD student with School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. He received his Bachelor of Engineering degree from Southwest Jiaotong University in 2020. He won the best Paper Award Nominee at ICIUS 2019.

Zhexiong Wan is currently a PhD student with School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. He received his Bachelor of Engineering degree from Northwestern Polytechnic University in 2019.

Yuchao Dai is currently a Professor with School of Electronics and Information at the Northwestern Polytechnical University (NPU). He received the B.E. degree, M.E degree and Ph.D. degree all in signal and information processing from Northwestern Polytechnical University, Xi'an, China, in 2005, 2008 and 2012, respectively. He was an ARC DECRA Fellow with the Research School of Engineering at the Australian National University, Canberra, Australia. His research interests include structure from motion, multi-view geometry, low-level computer vision, deep learning, compressive sensing and optimization. He won the Best Paper Award in IEEE CVPR 2012, the DSTO Best Fundamental Contribution to Image Processing Paper Prize at DICTA 2014, the Best Algorithm Prize in NRSFM Challenge at CVPR 2017, the Best Student Paper Prize at DICTA 2017, the Best Deep/Machine Learning Paper Prize at APSIPA ASC 2017, the Best Paper Award Nominee at IEEE CVPR 2020. He served as Area Chair in CVPR, ICCV, ACM MM, ACCV, WACV and etc.

Xin Yu received his B.S. degree in Electronic Engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2009, and received his Ph.D. degree in the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2015. He also received a Ph.D. degree in the College of Engineering and Computer Science, Australian National University, Canberra, Australia, in 2019. He is currently a lecturer in University of Technology Sydney. His interests include computer vision and image processing.