Self-supervised Non-uniform Kernel Estimation with Flow-based Motion Prior for Blind Image Deblurring

Zhenxuan Fang[1] Fangfang Wu[1*] Weisheng Dong[1] Xin Li[2] Jinjian Wu[1] Guangming Shi[1]
[1]Xidian University [2]West Virginia University
zxfang@stu.xidian.edu.cn wufangfang@xidian.edu.cn wsdong@mail.xidian.edu.cn
xin.li@mail.wvu.edu jinjian.wu@mail.xidian.edu.cn gmshi@xidian.edu.cn

Abstract

Many deep learning-based solutions to blind image deblurring estimate the blur representation and reconstruct the target image from its blurry observation. However, these methods suffer from severe performance degradation in real-world scenarios because they ignore important prior information about motion blur (e.g., real-world motion blur is diverse and spatially varying). Some methods have attempted to explicitly estimate non-uniform blur kernels by CNNs, but accurate estimation is still challenging due to the lack of ground truth about spatially varying blur kernels in real-world images. To address these issues, we propose to represent the field of motion blur kernels in a latent space by normalizing flows, and design CNNs to predict the latent codes instead of motion kernels. To further improve the accuracy and robustness of non-uniform kernel estimation, we introduce uncertainty learning into the process of estimating latent codes and propose a multi-scale kernel attention module to better integrate image features with estimated kernels. Extensive experimental results, especially on real-world blur datasets, demonstrate that our method achieves state-of-the-art results in terms of both subjective and objective quality as well as excellent generalization performance for non-uniform image deblurring. The code is available at https://see.xidian.edu.cn/faculty/wsdong/Projects/UFPNet.htm.

1. Introduction

Blind single image deblurring is a classic low-level vision problem that aims to recover the unknown sharp image from its observed blurry image without knowing the blur kernel. The uniform degradation model assumes that a blurry image is generated by a spatially invariant convolution process, which can be mathematically formulated as

$$\boldsymbol{y} = \mathbf{B}(\boldsymbol{x}, \boldsymbol{k}) + \boldsymbol{n}, \tag{1}$$

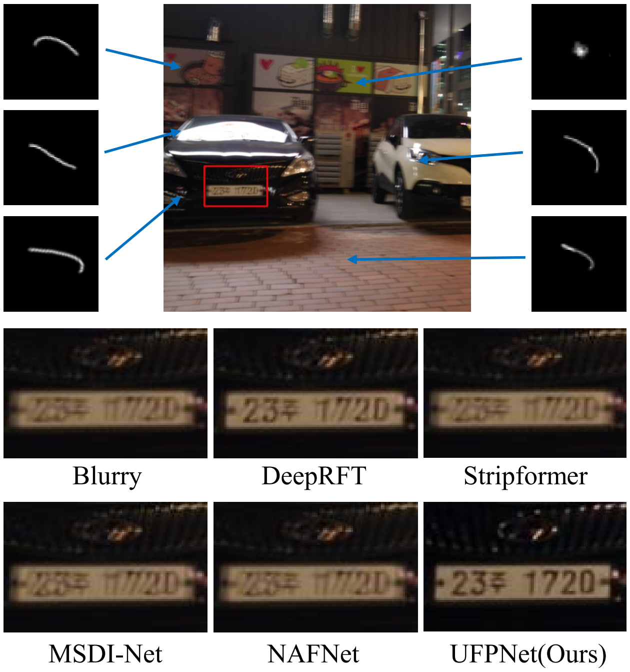

Blurry DeepRFT Stripformer
MSDI-Net NAFNet UFPNet(Ours)

Figure 1. The non-uniform kernel estimation and deblurring results of the proposed UFPNet on the RealBlur-J dataset.

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are sharp image and blurry image, respectively, $\mathbf{B}(\cdot, \boldsymbol{k})$ represents the blurring operator with the blur kernel $\boldsymbol{k}$ and $\boldsymbol{n}$ denotes the additive Gaussian noise. The simple case assumes the blur operation in Eq. (1) is uniform and the corresponding blur kernel is shift-invariant [11,43]. Several methods have been proposed to estimate the blur kernel and sharp image simultaneously [6,34,42]. However, in the real world, there are several factors that can cause blur degradation, such as camera shake and object movement. Although camera shake usually causes uniform and global background blurring, fast-moving objects often produce local blurring in the situation of a stationary background [50]. Therefore, the uniform blur in Eq. (1) is inappropriate for characterizing local blurring in the real world.

[*] Corresponding author

Traditional approaches to blind image deblurring first estimate the underlying blur kernels and then reconstruct the sharp image by iterative optimization [12, 28, 40, 44]. To constrain the solution space, both the image- and blur-related priors are exploited. In [29], the dark channel prior is used to estimate the blur kernel and reconstruct the sharp image. In [45], a novel extreme channel prior is proposed to facilitate the process of simultaneous image and kernel estimation. More recently, deep learning-based solutions have been proposed for blind image deblurring. Existing methods can be categorized into two classes. One class is to explicitly estimate the non-uniform blur kernel using convolutional neural networks (CNNs) [1, 2, 33, 37]. The other class of approaches is to use CNNs to directly reconstruct the original sharp image end-to-end without estimating the blur kernel [7, 19, 23, 26, 31, 46–48, 51]. DeepDeblur method [27] designs a multi-scale CNN to mimic conventional coarse-to-fine optimization and directly restores sharp images without assuming any restricted blur kernel model. SRN [38] proposes a scale-recurrent network and an encoder-decoder ResBlocks structure in each scale. Kupyn et al. propose DeblurGAN [19] and DeblurGAN-v2 [20] to reconstruct sharp images by adversarial training.

Unfortunately, both types of methods mentioned above have their fundamental limitations. First, since the characteristics of blur in real scenarios are complex, accurate estimation of non-uniform (i.e., spatially varying) blur kernel is challenging. For example, there exists an inevitable uncertainty in kernel estimation because a blurry image may have multiple kernel candidates due to its ill-posed nature. Therefore, incorrect blur kernels will lead to severe performance degradation in real-world image deblurring. Second, end-to-end methods ignore the information of motion prior, because the formation of image blur is usually associated with the motion trajectory of the camera and objects, which can be exploited for image deblurring effectively. The above observations inspire us to tackle the problem of blind image deblurring from a different perspective. The motivation for our work is threefold. On the one hand, since there is no ground truth of the blur kernel of real blur datasets, we attempt to simulate the non-uniform motion kernels to facilitate the kernel estimation in a *self-supervised* manner. On the other hand, we advocate a latent space approach to non-uniform blur kernel estimation, which is inspired by recent work on normalizing flows [13, 14, 16, 25]. Third, we introduce uncertainty learning to the process of estimating latent code, aiming to improve both the accuracy and robustness of non-uniform kernel estimation.

In this paper, we propose to model spatially varying motion blur prior by introducing normalizing flow and uncertainty learning in the latent space to kernel estimation. To address the issue of non-uniform blur that varies from pixel to pixel, we propose to represent the motion blur kernels in a latent space by normalizing flow and designing CNNs to predict spatially varying latent codes instead of motion kernels. This latent space approach can be interpreted as the generalization of the existing flow-based kernel prior (FKP) [24] from uniform to non-uniform by incorporating kernel generation from simulated random trajectories (e.g., DeblurGAN [19]). To further improve the accuracy and robustness of kernel estimation, we introduce uncertainty learning into the process of estimating latent codes and propose a multi-scale kernel attention module to better integrate image features with estimated kernels. The technical contributions of this paper are listed below.

- We propose to represent the non-uniform motion blur kernels in a latent space by normalizing flow. Our latent space approach allows CNNs to predict spatially varying latent codes rather than motion kernels. For the first time, we show how to estimate spatially varying motion blur on a pixel-by-pixel basis.
- To further improve performance and robustness, we introduce uncertainty learning to the latent code estimation process. The network learns the variance of the latent code to quantify the corresponding uncertainty, which leads to a more accurate prediction than the deterministic model.
- We propose a novel multi-scale kernel attention module to integrate image features and kernel information, which can be plugged into encoder-decoder architectures to incorporate the estimated kernels with the deblurring network.
- In view of the lack of ground truth about the non-uniform motion kernel in real-world images, we tackle the training set generation in a self-supervised manner. Extensive experimental results on benchmark datasets show that the proposed method significantly outperforms existing state-of-the-art methods and demonstrated excellent generalization performance from GoPro to other real-world blur datasets.

2. Related Work

2.1. Kernel Estimation in Image Deblurring

For blind image deblurring, early works use hand-crafted designed priors to constrain the solution space of blur, including total variation [3], heavy-tailed gradient prior [34], hyper-Laplacian prior [18] and l_0-norm gradient prior [44]. In the past decade, numerous deep models have been proposed and have achieved significant success. There are several methods that try to estimate blur kernels explicitly. [37] propose to predict the probabilistic distribution of the motion blur kernel at the patch level by CNNs. The authors of [33] perform kernel estimation by division in a Fourier space from the extracted deep features. In [10], the authors train CNNs to estimate the spatially invariant blur kernel and embed it into an unfolding reconstruction network. [2] predicts the complex Fourier coefficients of a deconvolution filter to be applied to the input patch.

2.2. Blind Image Deblurring

Recently, the widely used approaches are to directly estimate the original sharp image from the given blurred image without explicitly estimating non-uniform blur kernels [7, 19, 23, 26, 27, 31, 38, 46–48, 51]. MIMO-UNet [7] revisits the coarse-to-fine scheme and presents a multi-input, multi-output U-net. MPRNet [47] proposes a novel multi-stage progressive architecture to generate contextually enriched and spatially accurate outputs. HINet [5] introduces instance normalization into a residual block and designs a half-instance normalization block to boost the performance. DeepRFT [26] presents a residual fast Fourier transform block to integrate low and high-frequency information. MSDI-Net [22] proposes to learn the degradation representations of blurry images and integrate them into neural networks. Stripformer [39] develops a transformer-based architecture that constructs horizontal and vertical tokens to reweight image features. NAFNet [4] propose a simple baseline network for image deblurring.

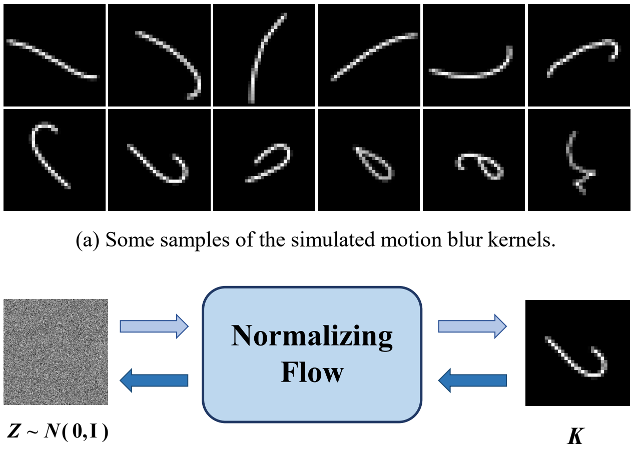
2.3. Normalizing Flow

Normalizing Flows [8, 9, 13, 14, 16, 17, 25, 30] are generative models which can deform the complex data distribution p_K to a simple distribution p_Z (usually a Gaussian distribution) by invertible neural networks. According to the change of variable formula [8], the concise negative log-likelihood loss function can be expressed as

$$\mathcal{L}(\mathbf{k};\boldsymbol{\theta}) = -\log p_Z\left(f_{\boldsymbol{\theta}}(\mathbf{k})\right) - \log\left|\det\left(\frac{\partial f_{\boldsymbol{\theta}}\left(\mathbf{k}\right)}{\partial \mathbf{k}}\right)\right|, \tag{2}$$

where $\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{k})}{\partial \mathbf{k}}$ is the Jacobian matrix of the flow model $f_{\boldsymbol{\theta}}$, and the parameter $\boldsymbol{\theta}$ is optimized by estimating maximum likelihood.

NICE [8] proposes a flow model by stacking nonlinear additive coupling and other transformation layers. The authors then upgrade the additive coupling to the affine coupling in RealNVP [9], which achieves better performance while retaining invertibility. Recently, normalizing flows have also been successfully applied in image restoration tasks such as super-resolution [24, 25]. Deflow [41] proposes a novel method based on conditional normalization of flow to learn degradations from unpaired data. FKP [24] proposes a Gaussian kernel prior to obtain the uniform blur kernel by optimizing the latent variable. Unlike FKP searching for a latent code, we propose to directly predict the latent code from the blurry image and use the estimated code to obtain the non-uniform blur field.


(a) Some samples of the simulated motion blur kernels.

(b) The illustration of the flow-based motion prior model.

Figure 2. The illustration of some motion blur kernels simulated from random trajectories and the normalizing flow learns a bijective mapping between the blur kernels and Gaussian distribution.

3. Proposed Method

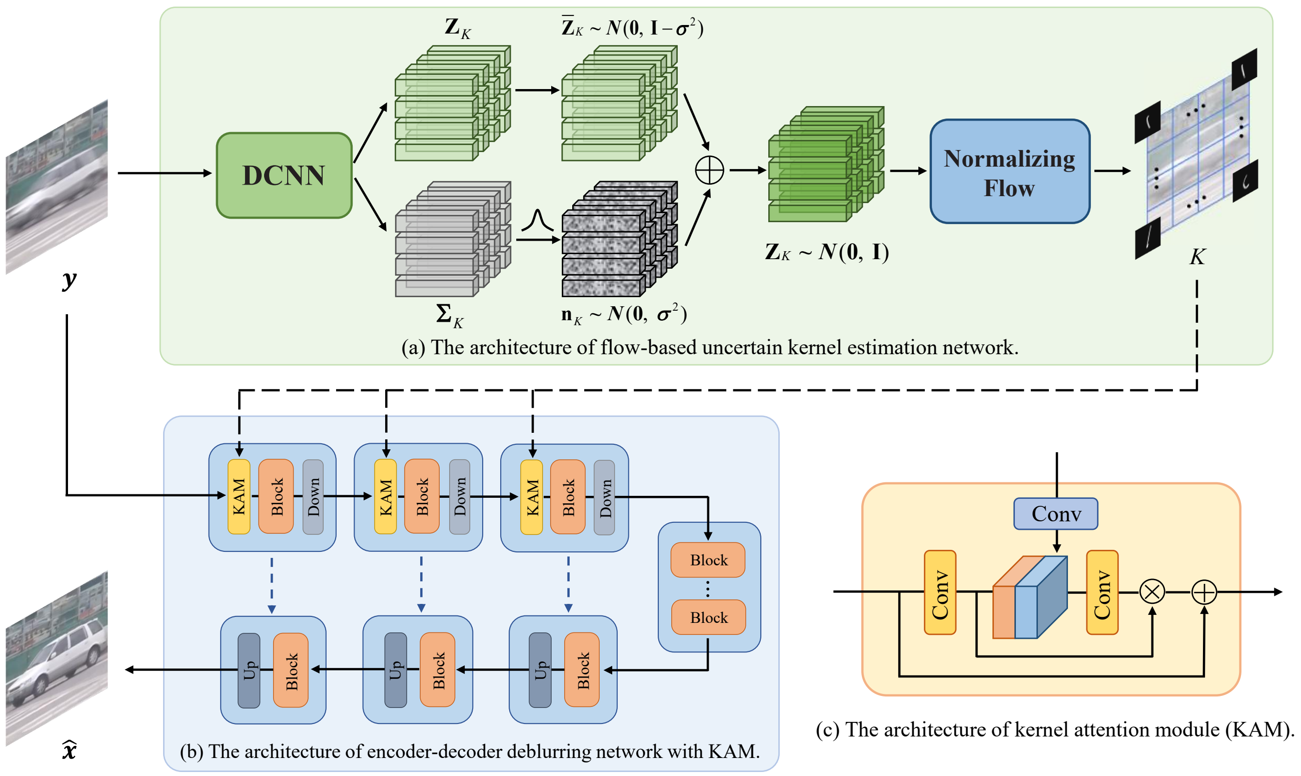
3.1. Self-supervised Kernel Estimation in Latent Space

There have been many approaches trying to estimate blur kernel from blurry image by CNNs [1, 2, 33, 37], they assume that the ground truth of the blur kernel is given or acquired through traditional optimization methods [21, 40]. To constrain the solution space of non-uniform blur, it is fundamental and important to exploit an effective prior [3, 18, 34, 44], and flow-based kernel prior [24] is a learning-based kernel prior that is applicable for arbitrary blur kernel modeling. However, due to the complexity of blur characteristics and the lack of ground truth of the real blurry image, these kernel estimation methods are not practical in real scenarios. Therefore, we propose to represent the complex motion blur kernel distribution into a simple Gaussian distribution by a normalizing flow and estimate the blur kernel in the latent space.

Specifically, a bijective mapping can be established between the kernel instance $\mathbf{k} \in K$ and the corresponding latent variable $\mathbf{z} \in Z$ by a normalizing flow: $\mathbf{k} \leftrightarrow \mathbf{z}$, which can be mathematically expressed as

$$\mathbf{k} = f_{\boldsymbol{\theta}}(\mathbf{z}), \quad f_{\boldsymbol{\theta}}^{-1}(\mathbf{k}) = \mathbf{z}, \tag{3}$$

where $f_{\boldsymbol{\theta}}(\cdot)$ denotes the flow model with parameter $\boldsymbol{\theta}$ and $f_{\boldsymbol{\theta}}^{-1}(\cdot)$ represents the inverse process of the flow. Once the motion blur kernel samples are given, we can train an invertible flow model that can transform between the kernel and the Gaussian distribution by Eq. (2). For motion blur kernels, we adopt a kernel generation method proposed by [19], which simulates realistic and complex blur kernels from random trajectories. Some motion blur kernel samples



(a) The architecture of flow-based uncertain kernel estimation network.

(b) The architecture of encoder-decoder deblurring network with KAM.

(c) The architecture of kernel attention module (KAM).

Figure 3. Overview of the proposed UFPNet for blind image deblurring. The architecture of (a) the flow-based uncertain kernel estimation network, (b) the encoder-decoder deblurring network with kernel attention module, (c) the kernel attention module (KAM).

are visualized in Fig. 2(a), and the illustration of the flow-based motion prior is shown in Fig. 2(b), the architecture of the flow model is illustrated in the supplementary materials. With the trained normalizing flow model, we can predict the relatively simple latent code instead of estimating the complicated motion kernel in a spatially adaptive manner.

To overcome the problem of lacking ground truth of blur kernel, we propose to estimate the blur kernel in a self-supervised manner. We adopt the L_1 loss between the blurry image and the reblurred image by the estimated kernels, aiming at making the estimated blur kernel closer to the real situation, and the loss function of the kernel estimation (KE) network can be expressed as

$$\mathcal{L}_{KE} = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n \otimes f_{\boldsymbol{\theta}}[\mathbf{G}(\boldsymbol{y}_n)] - \boldsymbol{y}_n\|_1 \, , \tag{4}$$

where $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ denotes the n-th sharp and blurry image pair, N denotes the total number of training samples, $\mathbf{G}(\boldsymbol{y}_n)$ denotes the latent codes estimated from $\boldsymbol{y}_n$ by deep network $\mathbf{G}$, $f_{\boldsymbol{\theta}}[\mathbf{G}(\boldsymbol{y}_n)]$ are the non-uniform blur kernels decoded by normalizing flow, $\otimes$ denotes the blur operation.

3.2. Uncertainty Learning in Kernel Estimation

In real scenarios, the characteristics of motion kernels are complex and spatially varying, making accurate estimation of blur kernels difficult. To improve the performance and robustness of the prediction results, we introduce uncertainty learning into the blur kernel estimation process. As shown in Fig. 3 (a), the kernel estimation network first takes blurry image $\boldsymbol{y} \in \mathbb{R}^{C\times H\times W}$ as input and predicts the normalized latent code $\boldsymbol{z}_i \in \mathbb{R}^{L\times 1\times 1}$ for each pixel i, where $L = k^2$ denotes the size of the blur kernel and $\boldsymbol{z}_i$ is of standard normal distribution. Then the latent code of the entire image is represented as $\boldsymbol{Z}_K \in \mathbb{R}^{L\times H\times W}$. Meanwhile, the standard deviation of the latent code is predicted simultaneously, denoted as $\boldsymbol{\sigma}_i \in \mathbb{R}^{L\times 1\times 1}$ of each pixel and $\boldsymbol{\Sigma}_K \in \mathbb{R}^{L\times H\times W}$ of the entire image. Then the uncertain component $\boldsymbol{n}_i$ of each latent code $\boldsymbol{z}_i$ is obtained by Gaussian resampling:

$$\boldsymbol{n}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_i^2), \tag{5}$$

where $\boldsymbol{n}_i \in \mathbb{R}^{L\times 1\times 1}$ has the same size as $\boldsymbol{z}_i$. Then we transform the standard deviation of $\boldsymbol{z}_i$ via $\bar{\boldsymbol{z}}_i = \sqrt{1-\boldsymbol{\sigma}_i^2}\boldsymbol{z}_i$ so that $\bar{\boldsymbol{z}}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}-\boldsymbol{\sigma}_i^2)$. With the assumption that $\bar{\boldsymbol{z}}_i$ and $\boldsymbol{n}_i$ are independent, the final latent code can be regarded as the sum of them, and the result still satisfies the standard normal distribution:

$$\hat{\boldsymbol{z}}_i = \bar{\boldsymbol{z}}_i + \boldsymbol{n}_i, \quad \hat{\boldsymbol{z}}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

where $\bar{\boldsymbol{z}}_i$ denotes the identity component of the latent code and $\boldsymbol{n}_i$ is the uncertain component. Since $\hat{\boldsymbol{z}}_i$ is corrupted by random noise $\boldsymbol{n}_i$ during the training period, $\hat{\boldsymbol{z}}_i$ is no longer a deterministic point embedding, thus the model becomes more robust to the estimation error and can improve accuracy. Then the blur kernel of each pixel can be decoded by the pre-trained normalizing flow model in Section 3.1 using $\boldsymbol{k}_i = f_{\boldsymbol{\theta}}(\hat{\boldsymbol{z}}_i)$, where $\boldsymbol{k}_i$ denotes the final estimated blur kernel at pixel i, $f_{\boldsymbol{\theta}}(\cdot)$ is the pre-trained flow model.

3.3. Uncertain Flow-based Prior Network

Multi-scale Kernel Attention Module. In order for the deblurring network to utilize the information from the estimated blur kernel sufficiently, we propose a novel kernel attention module (KAM). As illustrated in Fig. 3 (c), let $\boldsymbol{X} \in \mathbb{R}^{C\times H\times W}$ be the input feature maps of the kernel attention module, we first process the feature maps with a convolution layer (Conv). Then the non-uniform kernel at each pixel $\boldsymbol{K} \in \mathbb{R}^{L\times H\times W}$ is also input to a convolution layer. And the kernel attention map is obtained by

$$\boldsymbol{F}_{att} = \text{Conv3}(\text{concat}[\text{Conv1}(\boldsymbol{X}), \text{Conv2}(\boldsymbol{K})]), \quad (7)$$

Then the output feature maps are expressed as $f(\boldsymbol{X}) = \boldsymbol{F}_{att} \odot \text{Conv1}(\boldsymbol{X}) + \boldsymbol{X}$, where $\odot$ denotes element-wise multiplication. Since the image feature size of the encoder in network is gradually decreasing, our kernel attention module at different depths adopts different convolution strides to adapt to the corresponding feature size. In this way, the information of the blur kernel can be integrated on the multi-scale feature maps. With the help of the proposed kernel attention module, image features will pay different attention to areas with different degrees of blurring, therefore achieving better results on non-uniform deblurring.

Overall Framework. The overall framework of the proposed uncertain flow-based prior network (UFPNet) is illustrated in Fig. 3. The blurry image is first input to the kernel estimation network and obtains the estimated blur kernel of each pixel. Then the kernels are integrated into the deblurring network using the kernel attention module. As illustrated in Fig. 3 (b), the kernel attention module is plugged into the front of each encoder. Without loss of generality, we apply our kernel estimation network and KAM to the NAFNet method [4], which is a simple and effective U-Net architecture for image restoration.

The training process of the entire network consists of three stages: (I) Pre-train the normalizing flow model to represent the motion blur kernel into a Gaussian distribution by Eq. (2); (II) The self-supervise loss of Eq. (4) is adopted to pre-train the kernel estimation network; (III) The PSNR loss [4, 5] is used to train the deblurring network, meanwhile, we use the reblur loss which can be expressed as

$$\mathcal{L}_{reblur} = \frac{1}{N}\sum_{n=1}^{N} \left\|\mathcal{F}(\boldsymbol{y}_n) \otimes \mathcal{K}(\boldsymbol{y}_n) - \boldsymbol{y}_n\right\|_1, \quad (8)$$

where $\mathcal{F}(\boldsymbol{y}_n)$ is the reconstructed image and $\mathcal{K}(\boldsymbol{y}_n)$ denotes the blur kernels obtained by the pre-trained kernel estimation network. The total reconstruction loss is described as $\mathcal{L}_{recon} = \mathcal{L}_{PSNR} + \lambda\mathcal{L}_{reblur}$, where λ is set to 0.01.

4. Experiments

4.1. Datasets and Implementation Details

Following previous state-of-the-art blind image deblurring methods [4, 22, 39], we train the proposed UFPNet on the GoPro dataset [27], which consists of 2,103 pairs of blurry and sharp images for training. For evaluation, we test our UFPNet on GoPro [27], HIDE [35] and RealBlur [32] testsets. The GoPro dataset includes 1,111 test images, the HIDE dataset provides 2,025 images for testing. The RealBlur dataset has 3,758 blurry and sharp pairs for training and 980 images for testing. We also train and test on RealBlur datasets following [39].

We adopt the training settings used in NAFNet [4], our model is trained with Adam optimizer [15] ($\beta_1 = 0.9$ and $\beta_2 = 0.9$) for a total of 400K iterations with the initial learning rate 0.001 with the cosine annealing schedule. The training patch size is 256×256 and the batch size is 64. We implement the proposed method by PyTorch.

4.2. Comparison with State-of-the-Art Methods

We have compared our method with several deblurring methods, including [4, 5, 7, 19, 20, 22, 26, 27, 31, 36, 38, 39, 47–49]. Results are directly cited from the original papers or generated by the official model released by the authors.

Quantitative comparison. The PSNR and SSIM results of the test methods for single image deblurring are reported in Table 1, the proposed UFPNet outperforms all comparison methods on each test set. Our method achieves $0.37dB$ improvement in terms of PSNR over the existing best-performing method NAFNet [4] on GoPro dataset. We also test UFPNet on the HIDE dataset, which is a human-aware motion image dataset. To demonstrate the generalization property and effectiveness of the flow-based motion prior, we further evaluated our method on the RealBlur-R and RealBlur-J datasets. As shown in Table 1, the results of our method on real blur images are significantly improved compared to other methods. Note that all the models mentioned above are trained on the GoPro training set, demonstrating the excellent generalization performance of our method from GoPro to other real-world blur datasets. We also train and test on RealBlur datasets, the results are shown in Table 3. The comparison on computational complexity in terms of MACs (G) are reported in Table 2.

Visual comparison. We have compared the deblurring visualization results produced by different methods in Fig. 1, 4, 5 and 6. From Fig. 4, we can see that the proposed method can reconstruct more high-frequency textures and

Method	GoPro		HIDE		RealBlur-R		RealBlur-J		Params
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	(M)
DeepDeblur [27]	29.23	0.916	N/A	N/A	32.51	0.841	27.87	0.827	11.7
SRN [38]	30.26	0.934	28.36	0.915	35.66	0.947	28.56	0.867	6.8
DeblurGAN [19]	28.70	0.858	24.51	0.871	33.79	0.903	27.97	0.834	N/A
DeblurGAN-v2 [20]	29.55	0.934	26.61	0.875	35.26	0.944	28.70	0.866	60.9
DBGAN [49]	31.10	0.942	28.94	0.915	N/A	N/A	N/A	N/A	11.6
DMPHN [48]	31.20	0.945	29.09	0.924	35.70	0.948	28.42	0.860	21.7
MT-RNN [31]	31.15	0.945	29.15	0.918	N/A	N/A	N/A	N/A	2.6
SAPHN [36]	31.85	0.948	29.98	0.930	N/A	N/A	N/A	N/A	23.0
MIMO-UNet [7]	32.45	0.957	29.99	0.930	35.54	0.947	27.63	0.837	16.1
MPRNet [47]	32.66	0.959	30.96	0.939	35.99	0.952	28.70	0.873	20.1
HINet [5]	32.71	0.959	30.32	0.932	35.75	0.949	28.17	0.849	88.7
DeepRFT [26]	33.23	0.963	31.42	0.944	35.86	0.950	28.97	0.884	23.0
Stripformer [39]	33.08	0.962	31.03	0.940	36.07	0.952	28.82	0.876	20.0
MSDI-Net [22]	33.28	0.964	31.02	0.940	35.88	0.952	28.59	0.869	135.4
NAFNet [4]	33.69	0.967	31.32	0.943	35.50	0.953	28.32	0.857	67.8
UFPNet (ours)	**34.06**	**0.968**	**31.74**	**0.947**	**36.25**	**0.953**	**29.87**	**0.884**	80.3

Table 1. The comparison results on the benchmark datasets, the models are trained only on the GoPro dataset.

Method	MIMO-UNet [7]	DeepRFT [26]	MPRNet [47]	NAFNet [4]	MSDI-Net [22]	UFPNet (ours)
MACs (G)	1235.3	187.0	778.2	65.0	336.4	243.3

Table 2. The comparison on computational complexity in terms of MACs (G), when the input size is 256×256.

Method	RealBlur-R		RealBlur-J	
	PSNR	SSIM	PSNR	SSIM
DeblurGAN-v2 [20]	36.44	0.935	29.69	0.870
SRN [38]	38.65	0.965	31.38	0.909
MIMO-UNet [7]	N/A	N/A	31.92	0.919
MPRNet [47]	39.31	0.972	31.76	0.922
DeepRFT [26]	39.84	0.972	32.19	0.931
Stripformer [39]	39.84	0.974	32.48	0.929
UFPNet (ours)	**40.61**	**0.974**	**33.35**	**0.934**

Table 3. The comparison results on RealBlur datasets. The models are trained and tested on the corresponding datasets.

sharper edges than other methods on GoPro dataset. In Fig 5, it is obvious that our method can restore more natural body characteristics on HIDE dataset. As can be seen in Fig. 6, our method achieves good results in removing motion blur in real blur images.

4.3. Ablation Studies

We conduct sufficient ablation studies to analyze the proposed method, including the effectiveness of flow-based motion prior, the effectiveness of uncertainty learning and the effectiveness of the proposed kernel estimation module.

Effectiveness of Flow-based Motion Prior. To demonstrate the effectiveness of flow-based motion prior in kernel estimation, we remove the normalizing flow model of the kernel estimation network in Fig. 3 (a), this simplified network directly estimates the blur kernel instead of estimating the latent code. We also compare it with a traditional non-uniform kernel estimation method proposed by Whyte et al. [40]. To measure the accuracy of the estimated blur kernel, we first compare the PSNR and SSIM results between the original blurry image and the generated blurry image using the predicted blur kernels by these kernel estimation methods. As shown in Table 4, using the network to estimate blur kernel (denoted as the baseline model) has higher results than the traditional method, and adding normalizing flow to the network can further improve the accuracy of estimation. Then we integrate the estimated blur kernels by these blur kernel estimation methods into the deblurring network as shown in Fig. 3, and compare the deblurring results in Table 5, which indicates that the closer estimated blur kernel is to the real situation, the better deblurring results we can get.

Effectiveness of Uncertainty Learning. To demonstrate the effectiveness of uncertainty learning (UL), the kernel estimation network is modified into a deterministic model by removing the variance branch and the random

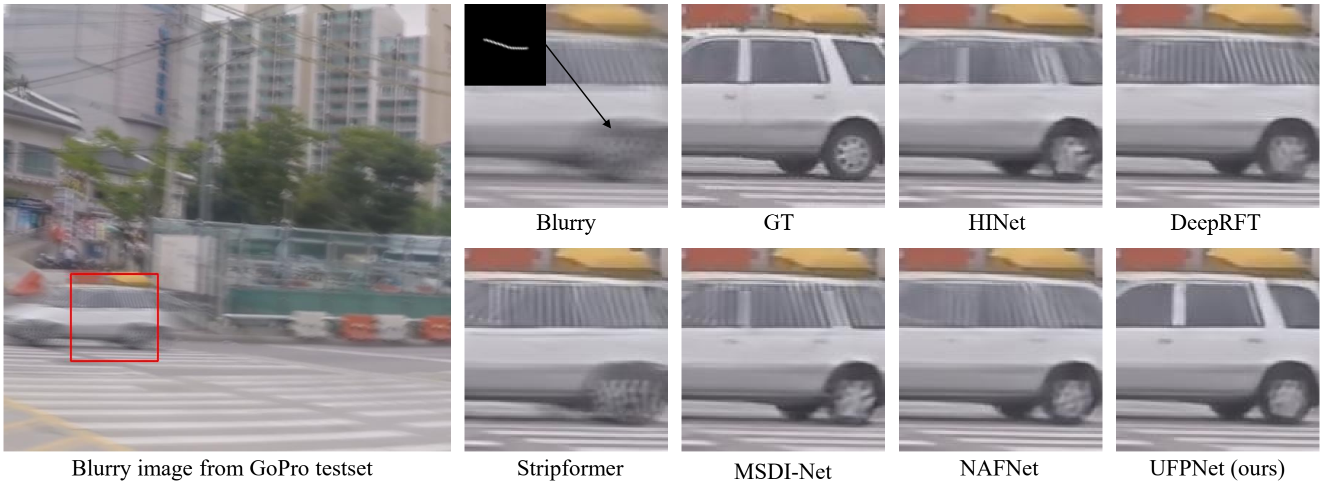


Figure 4. Visual comparisons on the GoPro dataset. From left to right: blurry image, ground-truth, results by HINet [5], DeepRFT [26], Stripformer [39], MSDI-Net [22], NAFNet [4] and UFPNet (ours). The estimated kernel at the indicated pixel is illustrated on the left-top.

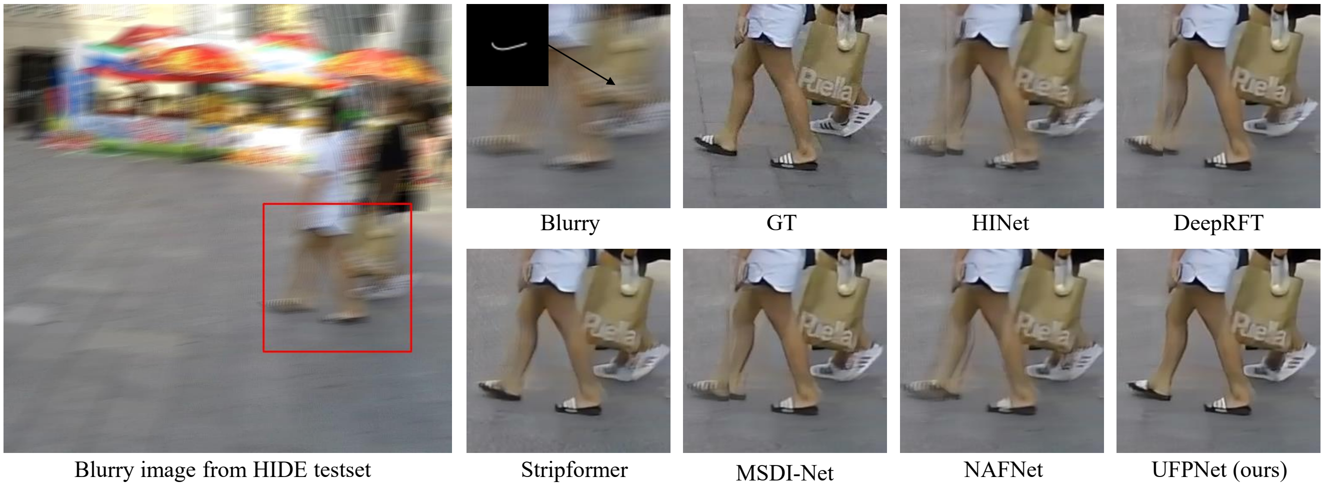


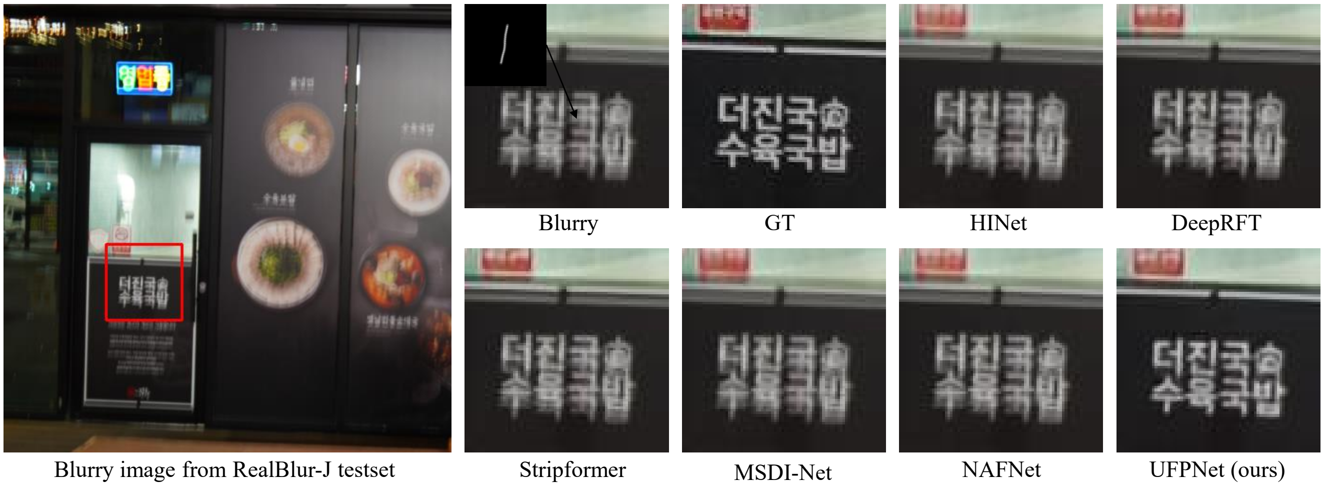
Figure 5. Visual comparisons on the HIDE dataset. From left to right: blurry image, ground-truth, results by HINet [5], DeepRFT [26], Stripformer [39], MSDI-Net [22], NAFNet [4] and UFPNet (ours). The estimated kernel at the indicated pixel is illustrated on the left-top.

Whyte et al. [40]	Proposed KE-Net			PSNR	SSIM
	Baseline	Flow prior	UL		
✓				41.63	0.989
	✓			43.90	0.993
	✓	✓		44.56	0.994
	✓	✓	✓	**45.92**	**0.996**

Table 4. PSNR and SSIM results between the original blurry image and the generated blurry image using the predicted blur kernels, which are estimated by the method of [40] and the variants of the proposed kernel estimation network (KE-Net) (w/ or w/o flow prior and uncertainty learning) on GoPro dataset. The baseline model denotes that the kernel estimation network directly estimates the blur kernel instead of the latent code.

noise component. Similarly, we compare the reblurring results in Table 4. When uncertainty learning is introduced, the results of blur kernel estimation can be significantly improved. For the deblurring results, as can be seen in Table 5 that without any kernel estimation method, our method degrades into the original NAFNet method [4]. With the help of the estimated kernel by Whyte et al. [40], the deblur results slightly outperform the original method. And the introduction of uncertainty learning can not only help improve the accuracy of blur kernel estimation but also improve the performance of image deblurring.

Effectiveness of the Kernel Estimation Module. Since our flow-based blur kernel estimation network and the multi-scale kernel attention module can be plugged into encoder-decoder architectures easily, we have upgraded some other image deblurring networks to show the effectiveness of the proposed flow-based kernel estimation mod-


Blurry image from RealBlur-J testset

Blurry

GT

HINet

DeepRFT

Stripformer

MSDI-Net

NAFNet

UFPNet (ours)

Figure 6. Visual comparisons on the RealBlur-J dataset. From left to right: blurry image, ground-truth, results by HINet [5], DeepRFT [26], Stripformer [39], MSDI-Net [22], NAFNet [4] and UFPNet (ours). The estimated kernel at the indicated pixel is illustrated on the left-top.

Whyte et al. [40]	Proposed KE-Net			GoPro		HIDE		RealBlur-R		RealBlur-J	
	Baseline	Flow prior	UL	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
				33.69	0.967	31.32	0.943	35.50	0.951	28.32	0.857
✓				33.74	0.967	31.38	0.944	35.61	0.951	28.79	0.863
	✓			33.78	0.967	31.45	0.945	35.78	0.952	29.13	0.869
	✓	✓		33.83	0.967	31.53	0.946	35.91	0.952	29.32	0.872
	✓	✓	✓	**34.06**	**0.968**	**31.74**	**0.947**	**36.25**	**0.953**	**29.87**	**0.884**

Table 5. The ablation studies of image deblurring results using different kernel estimation methods on benchmark datasets. The first row of the results denotes end-to-end training without estimating the blur kernel.

ule in improving image deblurring performance, including MPRNet [47], MIMO-UNet [7] and NAFNet [4]. Similar to the proposed UFPNet, the multi-scale kernel attention module is embedded in the front of each encoder. For a fair comparison, we use the codes released by their authors to retrain the modified models. As shown in Table 6, after using our kernel estimation network to predict the non-uniform blur kernels and embedding them into deep networks, the deblurring results are significantly improved.

Method	KE	GoPro		HIDE	
		PSNR	SSIM	PSNR	SSIM
MIMO-UNet [7]	×	32.45	0.957	29.99	0.930
	✓	**32.83**	**0.959**	**30.16**	**0.931**
MPRNet [47]	×	32.66	0.959	30.96	0.939
	✓	**33.04**	**0.967**	**31.13**	**0.941**
NAFNet [4]	×	33.69	0.964	31.32	0.943
	✓	**34.06**	**0.968**	**31.74**	**0.947**

Table 6. The image deblurring results of several networks w/ or w/o our kernel estimation (KE) module, which includes the blur kernel estimation network and the multi-scale kernel attention module. And using our KE module can bring significant improvement.

5. Conclusions

In this paper, we propose to represent the motion blur kernels in a latent space by a normalizing flow and designing CNNs to predict spatially varying latent codes instead of motion kernels. To further improve the accuracy and robustness of kernel estimation, we introduce uncertainty learning into the process of estimating latent codes and propose a multi-scale kernel attention module to better integrate image features with estimated kernels. To address the issue of the lack of ground truth about the non-uniform motion kernel in real-world images, we tackle the training set generation in a self-supervised manner. Extensive experimental results on benchmark datasets show that the proposed method significantly outperforms existing state-of-the-art methods and demonstrated excellent generalization performance from GoPro to other real-world blur datasets.

Acknowledgments. This work was supported in part by the Natural Science Foundation of China under Grant 61991451 and Grant 61836008. Xin Li's work is partially supported by the NSF under grant IIS-2114664, CMMI-2146076, and the WV Higher Education Policy Commission Grant (HEPC.dsr.23.7).

References

[1] Yuval Bahat, Netalee Efrat, and Michal Irani. Non-uniform blind deblurring by reblurring. In *Proceedings of the IEEE international conference on computer vision*, pages 3286–3294, 2017. 2, 3

[2] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *European conference on computer vision*, pages 221–235. Springer, 2016. 2, 3

[3] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE transactions on Image Processing*, 7(3):370–375, 1998. 2, 3

[4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 17–33. Springer, 2022. 3, 5, 6, 7, 8

[5] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 182–192, 2021. 3, 5, 6, 7, 8

[6] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–8. 2009. 1

[7] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021. 2, 3, 5, 6, 8

[8] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 3

[9] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 3

[10] Zhenxuan Fang, Weisheng Dong, Xin Li, Jinjian Wu, Leida Li, and Guangming Shi. Uncertainty learning in kernel estimation for multi-stage blind image super-resolution. In *European Conference on Computer Vision*, pages 144–161. Springer, 2022. 2

[11] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *Acm Siggraph 2006 Papers*, pages 787–794. 2006. 1

[12] Amit Goldstein and Raanan Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision*, pages 622–635. Springer, 2012. 2

[13] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR, 2018. 2, 3

[14] Priyank Jaini, Kira A Selby, and Yaoliang Yu. Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, pages 3009–3018. PMLR, 2019. 2, 3

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018. 2, 3

[17] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016. 3

[18] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in neural information processing systems*, 22, 2009. 2, 3

[19] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018. 2, 3, 5, 6

[20] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8878–8887, 2019. 2, 5, 6

[21] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971. IEEE, 2009. 3

[22] Dasong Li, Yi Zhang, Ka Chun Cheung, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. Learning degradation representations for image deblurring. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 3, 5, 6, 7, 8

[23] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 2, 3

[24] Jingyun Liang, Kai Zhang, Shuhang Gu, Luc Van Gool, and Radu Timofte. Flow-based kernel prior with application to blind super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10601–10610, 2021. 2, 3

[25] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *European conference on computer vision*, pages 715–732. Springer, 2020. 2, 3

[26] Xintian Mao, Yiming Liu, Wei Shen, Qingli Li, and Yan Wang. Deep residual fourier transformation for single image deblurring. *arXiv preprint arXiv:2111.11745*, 2021. 2, 3, 5, 6, 7, 8

[27] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 2, 3, 5, 6

[28] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2908, 2014. 2

[29] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel

prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1628–1636, 2016. 2

[30] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017. 3

[31] Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *European Conference on Computer Vision*, pages 327–343. Springer, 2020. 2, 3, 5, 6

[32] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *European Conference on Computer Vision*, pages 184–201. Springer, 2020. 5

[33] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1439–1451, 2015. 2, 3

[34] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *Acm transactions on graphics (tog)*, 27(3):1–10, 2008. 1, 2, 3

[35] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5572–5581, 2019. 5

[36] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3606–3615, 2020. 5, 6

[37] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 769–777, 2015. 2, 3

[38] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8174–8182, 2018. 2, 3, 5, 6

[39] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, and Chia-Wen Lin. Stripformer: Strip transformer for fast image deblurring. *arXiv preprint arXiv:2204.04627*, 2022. 3, 5, 6, 7, 8

[40] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce. Non-uniform deblurring for shaken images. *International journal of computer vision*, 98(2):168–186, 2012. 2, 3, 6, 7, 8

[41] Valentin Wolf, Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deflow: Learning complex image degradations from unpaired data with conditional flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 94–103, 2021. 3

[42] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *European conference on computer vision*, pages 157–170. Springer, 2010. 1

[43] Li Xu, Jimmy S Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. *Advances in neural information processing systems*, 27, 2014. 1

[44] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1107–1114, 2013. 2, 3

[45] Yanyang Yan, Wenqi Ren, Yuanfang Guo, Rui Wang, and Xiaochun Cao. Image deblurring via extreme channels prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4003–4011, 2017. 2

[46] Yuan Yuan, Wei Su, and Dandan Ma. Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3555–3564, 2020. 2, 3

[47] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021. 2, 3, 5, 6, 8

[48] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5978–5986, 2019. 2, 3, 5, 6

[49] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2737–2746, 2020. 5, 6

[50] Kaihao Zhang, Wenqi Ren, Wenhan Luo, Wei-Sheng Lai, Björn Stenger, Ming-Hsuan Yang, and Hongdong Li. Deep image deblurring: A survey. *International Journal of Computer Vision*, 130(9):2103–2130, 2022. 1

[51] Wenbin Zou, Mingchao Jiang, Yunchen Zhang, Liang Chen, Zhiyong Lu, and Yi Wu. Sdwnet: A straight dilated network with wavelet transformation for image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1895–1904, 2021. 2, 3