Learning Deblurring Texture Prior from Unpaired Data with Diffusion Model

Chengxu Liu[1] Lu Qi[2,3] Jinshan Pan[4] Xueming Qian[1] Ming-Hsuan Yang[5]
[1]School of Software Engineering, Xi'an Jiaotong University [2]Wuhan University
[3]Insta360 [4]Nanjing University of Science and Technology [5]University of California, Merced

Abstract

Since acquiring large amounts of realistic blurry-sharp image pairs is difficult and expensive, learning blind image deblurring from unpaired data is a more practical and promising solution. Unfortunately, dominant approaches rely heavily on adversarial learning to bridge the gap from blurry domains to sharp domains, ignoring the complex and unpredictable nature of real-world blur patterns. In this paper, we propose a novel diffusion model (DM)-based framework, dubbed TP-Diff, for image deblurring by learning spatially varying texture prior from unpaired data. In particular, TP-Diff performs DM to generate the prior knowledge that aids in recovering the textures of blurry images. To implement this, we propose a Texture Prior Encoder (TPE) that introduces a memory mechanism to represent the image textures and provides supervision for DM training. To fully exploit the generated texture priors, we present the Texture Transfer Transformer layer (TTformer), in which a novel Filter-Modulated Multi-head Self-Attention (FM-MSA) efficiently removes spatially varying blurring through adaptive filtering. Furthermore, we implement a wavelet-based adversarial loss to preserve high-frequency texture details. Extensive evaluations show that TP-Diff provides a promising unsupervised deblurring solution and outperforms SOTA methods in widely-used benchmarks.

1. Introduction

Image deblurring has been a topic of significant interest. The diversity and spatial variability of blurs pose significant challenges for developing effective solutions. Early conventional approaches [1, 16, 32, 33, 54] aim to find the prior distribution of sharp images and maximize the posterior probability, yet lack generalization ability.

Recent years have witnessed a growing interest in deep learning. Most approaches [2, 11, 22, 26] use synthetic paired data to supervise the DNN, achieving superior performance. However, the real-world blur is complex and unpredictable, making it infeasible to simulate real paired data manually. While recent studies [38, 62] have developed paired datasets using a dual-camera system, it is both expensive and time-consuming. In addition, it risks causing the model to overfit the specific characteristics of the camera used. Therefore, learning deblurring directly from unpaired blurry-sharp data presents a promising solution.

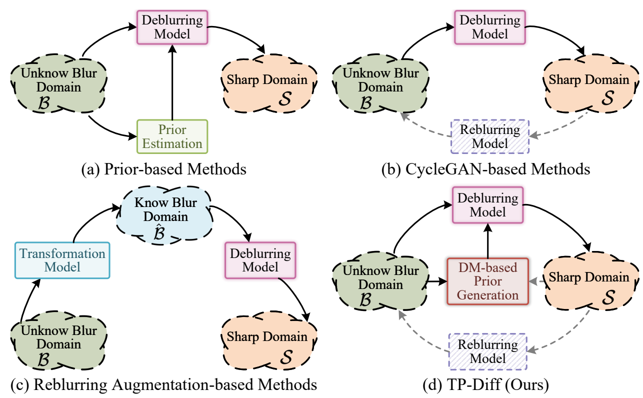

Figure 1. Illustration of previous unsupervised deblurring methods and our framework. (d) Our TP-Diff leverages the DMs to denoise pure Gaussian noise into spatially varying texture prior for deblurring tasks. The dotted line indicates use only during training.

Existing unsupervised deblurring methods can be categorized into three paradigms. 1) Prior-based methods [8, 15, 36, 45, 60] aim to train a sub-network that estimates prior knowledge for blur removal, as illustrated in Fig. 1(a). However, it is nearly impossible to find proper prior knowledge to model various blurs simultaneously. 2) Reblurring augmentation-based methods [35, 51] employ a transformation model to convert blurs from an unknown domain to a known one, thus reducing the difficulty of the deblurring process, as illustrated in Fig. 1(c). Although favorable results are achieved, these methods are limited by the ability of existing deblurring models and incur additional computational costs. 3) CycleGAN-based methods [3, 9, 27, 28, 61] construct blurry-sharp conversion cycles to learn the mapping between these two domains, as illustrated in Fig. 1(b). Unfortunately, these approaches overlook the spatial diversity of blur degrees and often tend to overfit a single blur template. These limitations hinder the advancement and effectiveness of unsupervised image deblurring.

Recently, diffusion models (DMs) have shown impressive performance in image generation [13, 14, 40, 43]. In image deblurring, most approaches use DMs directly for synthesizing sharp images, which eliminates blur but also introduces unpredictable artifacts [17, 37, 50]. In contrast, HiDiff [5] proposes using DMs to estimate a latent prior representation of sharp images for assisting the deblurring. Nevertheless, such prior is spatially out-of-order with a specific quantity and cannot be learned from unpaired data. Therefore, this inspires us to address the problem of *how to enable DMs to learn a spatially varying texture prior and exploit it in the unpaired deblurring task.*

To this end, we propose a novel diffusion model-based framework for unsupervised image deblurring (TP-Diff). The key insight is enabling the DM to learn the spatially varying texture prior from unpaired data and assist the deblurring process, as illustrated in Fig. 1(d). In particular, we follow [3, 61] to construct the cycle structure between sharp and blurry domains to learn the mapping relationships. The training process of TP-Diff involves two stages, as illustrated in Fig. 2(b). In the first stage, we use the proposed Texture Prior Encoder (TPE) to learn texture representations from extensive unpaired data and encode them to obtain texture priors in latent space. In addition, we propose a Texture Transfer Transformer (TTformer) layer within the deblurring network to employ the learned texture priors effectively. In the second stage, we freeze the parameters of TPE and train the DM to generate the texture prior using the TPE's output as ground truth. During inference, the denoising network produces the texture prior from pure Gaussian noise, which guides the deblurring network. Compared to generating sharp images directly, generating the texture prior in latent space requires fewer iterations [5, 53].

Our contributions are summarized as follows:

- We propose TP-Diff for unsupervised image deblurring, in which the DM can learn spatially varying texture prior efficiently for blur removal. To our knowledge, this is the first work to integrate DM into unpaired restoration tasks.
- We propose a TPE that introduces a memory mechanism to encode texture priors of blurry images in latent space.
- We propose a TTformer layer, where a Filter-Modulated Multi-head Self-Attention (FM-MSA) employs adaptive filtering to remove spatially varying blurs.
- We propose a wavelet-based adversarial loss that helps preserve high-frequency texture details during training.
- Extensive experiments demonstrate that TP-Diff outperforms SOTA methods on widely-used benchmarks.

2. Related Work

2.1. Deep Supervised Image Deblurring

Benefiting from the synthetic large-scale paired data, deep learning has achieved significant success in supervised image deblurring. Among them, some prior-free methods [2, 6, 22, 24, 30, 47, 57, 58] attempt to develop more robust models that directly learn to remove blur using multi-scale learning and supervision [2, 6, 57], attention mechanism [47, 58], and frequency domain learning [22, 30]. However, the shift-invariant of CNNs limits their effectiveness in handling spatially varying blurs.

To tackle this issue, prior-related methods [5, 11, 19, 26] learn the blur prior to guiding the deblurring network. Typically, UFPNet [11] and HiDiff [5] predict the prior representations through flow-based and diffusion-based models, respectively. Although achieving significant performance, they are infeasible to learn the prior from unpaired data. Significantly, such practice of predicting the prior from the inputs is treated as an inverse problem of blur patterns, providing inspiration for addressing spatially varying blurs.

2.2. Deep Unsupervised Image Deblurring

Unsupervised image deblurring is directly trained using unpaired data. The absence of pixel-level constraints makes it challenging. Existing approaches can be categorized into three paradigms: prior-based, reblurring augmentation-based, and CycleGAN-based methods. According to the distribution characteristics of sharp data, the prior-based methods [8, 36, 60] use maximum the posterior probability to estimate the prior for the deblurring process. However, it is almost impossible to deal with multiple blurs simultaneously by assuming prior knowledge. To avoid training unpaired data directly, reblurring augmentation-based methods [18, 35, 51] train an additional transformation model to convert the input from an unknown blur domain to a known blur domain [35] or generate more diverse blur data [51]. Nevertheless, these methods still rely on the pre-trained deblurring models that were trained on synthetic paired data.

To directly implement training based on unpaired data, CycleGAN-based methods [3, 9, 27, 28, 61] construct sharp-blurry-sharp and blurry-sharp-blurry conversion cycles to learn the mapping between sharp and blurry domains by adversarial learning [12]. The recent SEMGUD [3] proposes a self-enhancement deblurring strategy to progressively improve the generated pseudo-paired data and reconstructor, achieving advanced results. However, this self-enhancement strategy depends on a fully supervised model trained on synthetic paired data and overlooks the spatial diversity of realistic blurs. In our work, TP-Diff enables tailored removal of various blurs without using any paired data by learning texture prior in different regions.

2.3. Diffusion Model

Recently, diffusion model (DM) [13, 43] has been attracting increasing attention in low-level vision tasks, such as super-resolution [25, 55], denoising [23], inpainting [29], and so on. In image deblurring, some works [17, 37, 50] directly

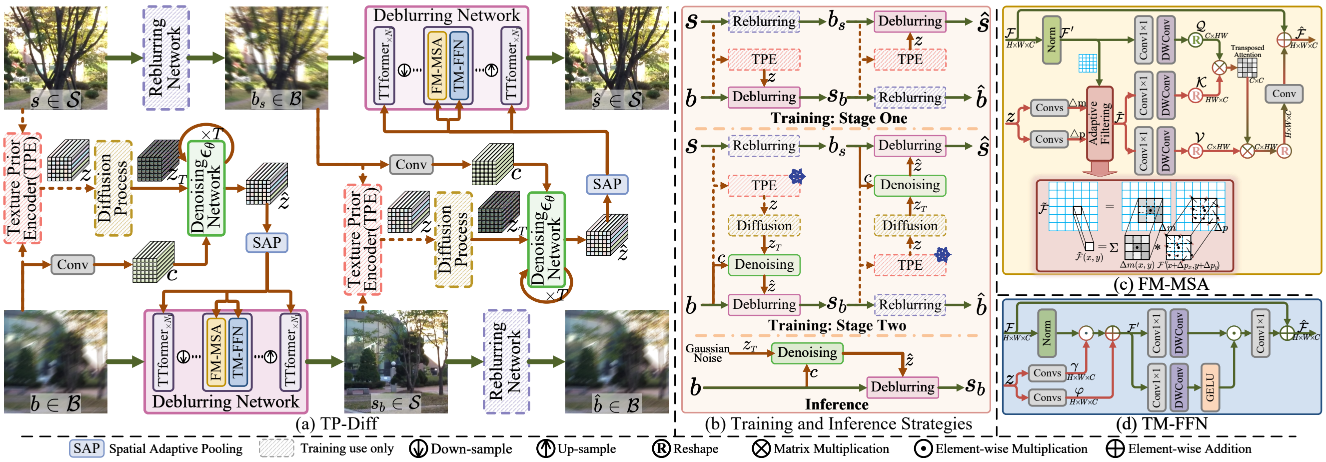

Figure 2. (a) Overview of TP-Diff. The Texture Prior Encoder (TPE) is used for extracting texture prior from unpaired data. The diffusion process and denoising network are used for adding noise and generating the texture prior, respectively. (b) Schematic of training and inference strategies. (c) Filter-Modulated Multi-head Self-Attention (FM-MSA) and (d) Transform-Modulated Feed-Forward Network (TM-FFN) form the Texture Transfer Transformer layer (TTformer), which is used to exploit the texture prior in the deblurring network.

use the DM to generate sharp images from Gaussian noise through a stochastic iterative denoising process. Although this practice can generate clear textures, it also introduces unpredictable artifacts. Other works [5, 53] use DM to estimate the latent prior representations of sharp images for assisting deblurring, yielding superior results while avoiding artifacts. Nevertheless, these methods ignore the spatial diversity of blur in real-world scenarios, and it is also unavailable to train with unpaired images. In this paper, we apply DM to learn spatially varying texture prior from unpaired data for deblurring, which is the first work to use DM for unpaired image reconstruction.

3. Methodology

3.1. Overall Architecture

As illustrated in Fig. 2(a), our TP-Diff mainly comprises four components: Texture Prior Encoder (TPE), Diffusion Model (DM), deblurring network, and reblurring network. Specifically, TPE is used to obtain the texture prior, which serves as the ground truth for DM training. DM consists of a diffusion process and a denoising network for adding noise and generating texture prior, respectively. The deblurring network and reblurring network together form the entire cycle structure designed for removing and synthesizing blur, respectively. Within the deblurring network, the proposed Texture Transfer Transformer (TTformer) layer leverages texture prior knowledge to effectively remove blur.

To train TP-Diff effectively, we follow established practices [5, 25] and divide the training process into two stages, as illustrated in Fig. 2(b). Taking the blurry image $b \in \mathcal{B}$ and unpaired sharp image $s \in \mathcal{S}$ as inputs. In stage one, we utilize the TPE to obtain spatially varying texture prior $z \in \mathbb{R}^{H\times W\times C}$ in latent space, and jointly train it and the entire cycle structure for blur removal. At this stage, the z output from TPE is directly fed into the deblurring network without involving the diffusion and denoising processes. In stage two, we freeze the parameters of TPE, and joint train DM and the entire cycle structure. This enables the denoising network to generate a more reliable spatially varying texture prior $\hat{z} \in \mathbb{R}^{H\times W\times C}$. At this stage, the z output from TPE first adds noise to output $z_T \in \mathbb{R}^{H\times W\times C}$ through the diffusion process, and then generates the prior $\hat{z}$ through multiple iterations of the denoising process. $H \times W \times C$ is the size of the prior. Please refer to the supplementary for more detailed algorithms and structures. Next, we describe the training strategy and the inference process in order.

3.2. Stage One: Latent Texture Prior Extraction

In this stage, we aim to obtain a high-quality texture prior that guides the blur removal. Specifically, we use TPE, which introduces a learnable memory bank, to learn texture representations from extensive unpaired data during training, thus extracting spatially varying texture prior z. Then, to fully leverage the texture prior and enhance the model capacity, we incorporate the TTformer at multiple scales within the deblurring network and fed z into each scale through spatial adaptive pooling (SAP). Below, we introduce the core components TPE and TTformer.

Texture Prior Encoder. As shown in Fig. 3, TPE consists of a texture-memory enhancement part and a texture-memory transfer part. In the enhancement part, we initialize a group of learnable memories, and extract tokens from texture-rich sharp images as reference texture templates to enhance the memories. In the transfer part, we search for similar tokens from the enhanced memories and use them to represent texture templates in the input blurry image.

Specifically, in the enhancement part, we first compress

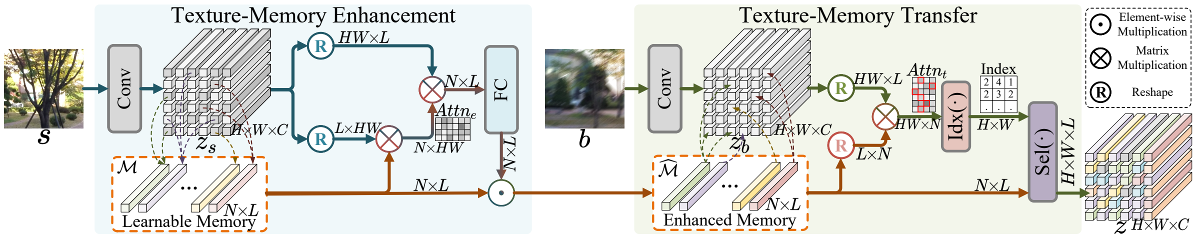

Figure 3. Structure of the Texture Prior Encoder (TPE), which serves to get spatially varying texture priors from the unpaired blurry-sharp input. $\mathrm{Idx}(\cdot)$ indicates that extracting the index with the highest score. $\mathrm{Sel}(\cdot)$ indicates that selecting the tokens according to the index.

sharp image s into a higher dimensional space by a convolutional layer and obtain the texture-rich tokens $z_s \in \mathbb{R}^{H\times W\times C}$. Then, we use z_s to enhance the introduced learnable memory $\mathcal{M} \in \mathbb{R}^{N\times L}$. N is the memory size, which is set to 256 in our model. L is the memory dimension, which is equal to C. In detail, we generate a corresponding attention map for the $\mathcal{M}$, formulated as:

$$Attn_e = \mathrm{SoftMax}(\mathcal{M} \otimes \mathrm{R}(z_s)), \tag{1}$$

where $Attn_e \in \mathbb{R}^{N\times HW}$ is the similarity of each memory with the texture-rich tokens. $\mathrm{SoftMax}(\cdot)$, $\mathrm{R}(\cdot)$, and $\otimes$ are the softmax function, reshape operation, and matrix multiplication, respectively. Next, we can compute enhanced memories $\widehat{\mathcal{M}}$ from all texture-rich tokens z_s, formulated as:

$$\widehat{\mathcal{M}} = \mathcal{M} \odot \mathrm{FC}(Attn_e \otimes \mathrm{R}(z_s)), \tag{2}$$

where $\odot$ denotes the element-wise multiplication. $\mathrm{FC}(\cdot)$ is the full connection layer used to refine the tokens after attention aggregation. This design allows memory to learn the valuable texture templates in all sharp data.

In the transfer part, we also compress blurry image b and obtain the blurry tokens $z_b \in \mathbb{R}^{H\times W\times C}$. Then, we search for similar tokens from the enhanced memories that have similar texture templates to those in the blurry input. The generated attention map $Attn_t \in \mathbb{R}^{HW\times N}$ is given by:

$$Attn_t = \mathrm{R}(z_b) \otimes \mathrm{R}(\widehat{\mathcal{M}}). \tag{3}$$

We select the enhanced memory with the highest attention map score for each blurry token and then aggregate them into the TPE's output z, formulated as:

$$z = \mathrm{R}(\mathrm{Sel}(\mathrm{Idx}(Attn_t); \widehat{\mathcal{M}})) \tag{4}$$

where $\mathrm{Idx}(\cdot)$ means extracting the index with the highest sorting score along the rows in the attention map. $\mathrm{Sel}(\cdot)$ means selecting the corresponding tokens from the enhanced memories for aggregation.

Different from the vanilla self-attention that takes a weighted sum of tokens, we use only the most relevant tokens to ensure that the selected memories represent regions with similar texture templates. Compared to directly learning in image space, the proposed prior in the feature space is more robust and stable. Moreover, the TPE is robust enough to benefit from different unpaired sharp images. We provide detailed analysis in Sec. 4.4.

TTformer. Existing Transformer-based restoration methods [22, 47, 58] achieve superior performance in supervised learning, but not explored in unsupervised feature learning. Therefore, we introduce the TTformer layer to efficiently integrate the obtained prior knowledge into the unsupervised deblurring. It consists of two components, the filter-modulated multi-head self-attention (FM-MSA) and the transform-modulated feed-forward network (TM-FFN).

Specifically, as shown in Fig. 2(c), for the input feature $\mathcal{F}$ and input prior z, FM-MSA exploits an adaptive filtering operation to effectively remove spatially varying blurs. In detail, we use two three-layer convolution $\mathrm{Convs}(\cdot)$ to predict the vertical and horizontal offsets $\Delta p \in \mathbb{R}^{H\times W\times 2K^2}$ and weight $\Delta m \in \mathbb{R}^{H\times W\times K^2}$ of the filter, formulated as:

$$\Delta p, \Delta m = \mathrm{Convs}(z), \mathrm{Convs}(z), \tag{5}$$

where K is the kernel size. Then, taking the pixel with coordinates (x, y) in the filtered feature $\tilde{\mathcal{F}} \in \mathbb{R}^{H\times W\times C}$ as example, the filtering process is formulated as:

$$\tilde{\mathcal{F}}(x,y) = \sum_{\Delta p(x,y)} \Big(\Delta m(x,y) * \mathcal{F}'\big(x+\Delta p_x, y+\Delta p_y\big)\Big), \tag{6}$$

where $\mathcal{F}' = \mathrm{Norm}(\mathcal{F})$ represents the normalized input feature $\mathcal{F} \in \mathbb{R}^{H\times W\times C}$. Subsequently, the visual tokens can be expressed as follows:

$$\begin{aligned} \mathcal{Q} &= \mathrm{R}(\mathrm{PDConv}(\mathcal{F}')), \\ \mathcal{K}, \mathcal{V} &= \mathrm{R}(\mathrm{PDConv}(\tilde{\mathcal{F}})), \mathrm{R}(\mathrm{PDConv}(\tilde{\mathcal{F}})), \end{aligned} \tag{7}$$

where $\mathrm{PDConv}(\cdot)$ are the 1×1 point-wise convolution and 3×3 depth-wise convolution. Finally, we follow existing work [58] to generate a transposed-attention map and perform dot-product with $\mathcal{V}$ to obtain the final output of FM-MSA $\hat{\mathcal{F}}$. This process can be described as:

$$\hat{\mathcal{F}} = \mathrm{Conv}(\mathrm{R}(\mathrm{SoftMax}(\frac{\mathcal{Q}\otimes\mathcal{K}}{\sqrt{C}}) \otimes \mathcal{V})) \oplus \mathcal{F}, \tag{8}$$

where C is the dimension of tokens. $\oplus$ denotes the element-wise addition. The design of adaptive filtering can fully utilize the texture prior and significantly improve the model's ability to recover various textures.

In addition, as shown in Fig. 2(d), we use TM-FFN, similar in [25, 53], to aggregate local features from the FM-MSA output. In detail, TM-FFN first uses texture prior z to get dynamic parameters $\gamma, \varphi \in \mathbb{R}^{H\times W\times C}$, which are then used to modulate the input feature $\mathcal{F}$ by:

$$\begin{aligned} \mathcal{F}' &= \mathrm{Norm}(\mathcal{F}) \odot \gamma \oplus \varphi, \\ \gamma, \varphi &= \mathrm{Convs}(z), \mathrm{Convs}(z), \end{aligned} \tag{9}$$

where $\mathcal{F}' \in \mathbb{R}^{H\times W\times C}$ is the modulated feature. Finally, we adopt the gating mechanism to enhance feature encoding and get the output $\hat{\mathcal{F}} \in \mathbb{R}^{H\times W\times C}$ of TM-FFN by:

$$\hat{\mathcal{F}} = \mathrm{Conv}(\mathrm{GELU}(\mathrm{PDConv}(\mathcal{F}')) \odot \mathrm{PDConv}(\mathcal{F}')) \oplus \mathcal{F}. \tag{10}$$

Optimization Objective. We aim to get the spatially varying texture prior z by TPE and jointly train it with the entire cycle structure. To achieve this, we follow existing works [3, 61], using the same adversarial loss $\mathcal{L}_{GAN}$ and cycle consistency loss $\mathcal{L}_{CYC}$ to supervise training. In addition, to preserve high-frequency texture detail as possible, we propose a wavelet-based adversarial loss, formulated as:

$$\begin{aligned} \mathcal{L}_{Wave} = & \mathbb{E}_{s\sim p_{\Phi(s)}}[\log D_S(\Phi(s))] \\ & + \mathbb{E}_{b\sim p_b}[\log(1 - D_S(\Phi(DN(b)))], \end{aligned} \tag{11}$$

where $\Phi(\cdot)$ denotes the extraction of high-frequency components using the wavelet transform. $DN(\cdot)$ is the deblurring network. $D_S(\cdot)$ is the discriminator and tries to maximize the distinction between deblurred images and sharp images. The full objective function for stage one $\mathcal{L}_{s1}$ is a weighted sum of the above losses, formulated as:

$$\mathcal{L}_{s1} = \lambda_{GAN}\mathcal{L}_{GAN} + \lambda_{CYC}\mathcal{L}_{CYC} + \lambda_{Wave}\mathcal{L}_{Wave}, \tag{12}$$

where the hyper-parameters λ_{GAN}, λ_{CYC}, and λ_{Wave} control the importance of each term.

3.3. Stage Two: Texture Prior Generation

In this stage, we jointly train the DM and the entire cycle structure so that the denoising network generates effective texture priors to enhance the deblurring network. Specifically, following the existing works [5, 25, 53], our DM consists of the forward diffusion process and the reverse denoising process. In the forward diffusion process, we first adopt the TPE, in which the parameters are frozen, to generate the texture prior $z \in \mathbb{R}^{H\times W\times C}$ as ground truth, and then add noise to z by the diffusion process and get the $z_T \in \mathbb{R}^{H\times W\times C}$ with the same distribution as the pure Gaussian noise. In the reverse denoising process, we use the denoise network to generate the reconstructed texture prior $\hat{z} \in \mathbb{R}^{H\times W\times C}$ using the extracted feature $c \in \mathbb{R}^{H\times W\times C}$ by a convolutional layer from blurry input as the condition and z as the target.

Diffusion Process. Following existing works [13, 43], we perform the forward Markov process starting from z, and gradually add Gaussian noise by T iterations as follows:

$$q(z_T \mid z) = \mathcal{N}(z_T; \sqrt{\bar{\alpha}_T}z, (1-\bar{\alpha}_T)\mathrm{I}), \tag{13}$$

where T is the total number of iteration steps. $\mathcal{N}(\cdot)$ denotes the Gaussian distribution. $\alpha = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$, where $t \in \{1, \ldots, T\}$, $\beta_{1:T} \in (0, 1)$, are hyper-parameters derived through iterative derivation with reparameterization [21] to control the amount of noise added at each step.

Denoising Process. Aiming to generate the texture prior $\hat{z}$ from Gaussian noise, we perform the Markov chain that runs backward from z_T to z, and gradually remove the noise by T iterations. In the inverse step from z_t to z_{t-1}:

$$\begin{aligned} q(z_{t-1} \mid z_t, z) &= \mathcal{N}(z_{t-1}; \mu_t(z_t, z), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\mathrm{I}), \\ \mu_t(z_t, z) &= \frac{1}{\sqrt{\alpha_t}}(z_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon), \end{aligned} \tag{14}$$

where ϵ represents the noise in z_t, which is the only uncertain variable that needs to be estimated at each step using the denoising network. Therefore, we use a neural network consisting of a series of stacked residual blocks, denoted as ϵ_θ, to estimate the noise with the condition c. Then, we further substitute ϵ_θ into Eq. (14) to obtain:

$$z_{t-1} = \frac{1}{\sqrt{\alpha_t}}(z_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(z_t, A_h, t)) + \sqrt{1-\alpha_t}\epsilon_t, \tag{15}$$

where $\epsilon_t \sim \mathcal{N}(0, \mathrm{I})$. $\epsilon_\theta(\mathrm{z}_t, c, t)$ is the noise estimated by the denoising network. By repeating the T times sampling iterations in Eq. (15), we can obtain the reconstructed texture prior $\hat{z}$. The purpose of using residual blocks as the denoising network is to ensure the same resolution of inputs and outputs while minimizing the model parameters. As shown in Fig. 2(a), the output $\hat{z}$ from the denoising network is finally utilized to guide the deblurring network. Our DM is more effective compared to other texture prior generation methods. We provide detailed analysis in Sec. 4.4.

Optimization Objective. Our objective is to joint train the denoising network ϵ_θ and the entire cycle structure. To achieve this, we additionally include the diffusion loss $\mathcal{L}_{diff}$ based on $\mathcal{L}_{s1}$ in Eq. (12), formulated as:

$$\mathcal{L}_{s2} = \mathcal{L}_{s1} + \lambda_{diff}\mathcal{L}_{diff}, \quad \mathcal{L}_{diff} = \|z - \hat{z}\|_1. \tag{16}$$

3.4. Inference

During inference, only given a blurry input image b, we first extract the blur feature c from the blurry input as the condition of denoising network. Then, we randomly sample pure Gaussian noise z_T. After T times denoising process in Eq. (15), denoising network generates the texture prior $\hat{z}$ using z_T and c. Finally, we feed $\hat{z}$ to the deblurring network consisting of TTformer to compute the deblurred result.

	Methods	GoPro [31]		HIDE [42]		RealBlur-R [38]		RealBlur-J [38]		Overhead	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	#Param(M)	Latency(s)
Supervised Training	MSDI-Net [24]	33.28	0.964	31.02	0.940	35.88	0.952	28.59	0.869	135.4	-
	NAFNet [2]	33.69	0.967	31.32	0.943	35.50	0.953	28.32	0.857	67.9	0.04
	icDPM [37]	33.20	0.963	30.96	0.938	-	-	28.81	0.872	52.0	-
	HI-Diff [5]	33.33	0.964	31.46	0.945	36.28	0.958	29.15	0.890	28.5	-
	UFPNet [11]	34.06	0.968	31.74	0.947	36.25	0.953	29.87	0.884	80.3	-
Unpaired Training	CycleGAN [63]	22.54	0.757	21.81	0.675	12.38	0.242	19.79	0.633	11.38	0.02
	UNIT [27]	22.58	0.778	22.21	0.702	28.94	0.717	24.55	0.755	11.38	0.03
	UID-GAN [28]	23.56	0.812	22.70	0.715	16.64	0.323	22.87	0.671	19.93	0.04
	LIR [9]	23.26	0.818	22.44	0.708	26.94	0.644	23.59	0.747	11.15	0.02
	U-GAT-IT [20]	20.09	0.703	20.65	0.632	13.64	0.323	16.84	0.604	278.96	-
	DCD-GAN [4]	21.99	0.772	23.19	0.777	17.27	0.362	20.99	0.671	11.38	0.02
	FCL-GAN [61]	24.59	0.831	23.43	0.782	28.37	0.663	25.35	0.736	24.56	0.01
	UVCGANv2 [46]	24.15	0.814	23.35	0.792	25.31	0.658	24.82	0.740	32.56	0.03
	UCL [49]	25.06	0.839	23.85	0.816	30.53	0.757	26.04	0.784	19.45	0.02
	TP-Diff	28.13	0.903	26.70	0.821	34.95	0.933	28.01	0.836	11.89	0.04
	SEMGUD [3]	29.06	0.927	27.64	0.892	35.51	0.946	28.01	0.844	67.9	0.04
	TP-Diff-*se*	30.16	0.934	28.21	0.909	35.32	0.947	28.03	0.843	11.89	0.04

Table 1. Quantitative comparison on the GoPro [31], HIDE [42], RealBlur-R [38], and RealBlur-J [38]. Latency(s) is computed on images with the size of 256×256 with an NVIDIA RTX 3090 GPU. Red and blue indicate the best and second best performance, respectively.

4. Experiments

4.1. Datasets and Metrics

We evaluate the our method on widely-used datasets: GoPro [31], HIDE [42], RealBlur [38], RB2V_Street [34], and RSBlur [39]. For fair comparisons, we follow existing works [3, 35] to split the training set of GoPro, RB2V_Street, and RSBlur datasets into separate blurry and sharp image parts to constitute unpaired blurry-sharp pairs for training. We conduct three sets of experiments: i) Using the GoPro training set for training and the test sets for GoPro, HIDE, RealBlur-R, and RealBlur-J for testing. ii) Using the RB2V_Street training set for training and its test set for testing. iii) Using the RSBlur training set for training and its test set for testing. We keep the same evaluation metrics of PSNR(dB) and SSIM as previous works [3, 35].

4.2. Implementation Details

We follow existing works [3, 63] to use the same discriminator and use the UNet [41] as the reblurring network. In the deblurring network, we set the number of TTformer as [4,6,6,4], the attention heads as [1,2,4,8], and the number of channels C as 48, and the kernel size K of adaptive filtering as 5. In the denoising network, we set the number of ResBlock to 5 and the iteration step T to 8. During training, the hyper-parameters λ_{GAN}, λ_{CYC}, λ_{Wave}, and λ_{diff} are set to 1, 0.1, 0.2, and 1, respectively. We use Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, learning rate is 1×10^{-4} for both training stages. The epoch number for each stage is 200. We set the batch size as 8 and the input patch size as 256×256 and augment the data with random horizontal and vertical flips. All experiments were based on PyTorch and trained on an NVIDIA RTX A6000 GPU.

Methods	RB2V_Street [34]	RSBlur [39]
Generalized Deblurring		
BSRGAN [59]	23.31/0.645	27.11/0.810
RSBlur [39]	23.42/0.603	26.98/0.798
NAFNet [2]	28.72/0.883	33.06/0.888
Restormer [58]	27.43/0.849	32.87/0.874
Reblurring Augmentation		
NAFNet [2]+Blur2Blur [35]	26.98/0.812	29.00/0.857
Restormer [58]+Blur2Blur [35]	25.97/0.750	28.89/0.850
Unpaired Training		
CycleGAN [63]	21.21/0.582	23.34/0.782
DualGAN [56]	21.02/0.556	22.78/0.704
UNIT [27]	20.53/0.519	26.49/0.734
UID-GAN [28]	22.03/0.564	26.48/0.713
LIR [9]	20.43/0.534	24.44/0.720
U-GAT-IT [20]	20.75/0.539	22.17/0.629
DCD-GAN [4]	21.20/0.537	25.90/0.704
FCL-GAN [61]	21.77/0.560	28.17/0.746
UVCGANv2 [46]	22.23/0.561	27.80/0.708
UCL [49]	21.38/0.556	26.81/0.717
TP-Diff	22.89/0.639	28.40/0.751

Table 2. Quantitative comparison (PSNR↑ and SSIM↑) on the RB2V_Street [34] and RSBlur [39] datasets.

4.3. Comparisons with State-of-the-art Methods

We evaluate our method against 16 SOTA models [2–5, 9, 11, 20, 24, 27, 28, 35, 37, 46, 49, 61, 63]. For fair comparisons, we obtain the performance from their original

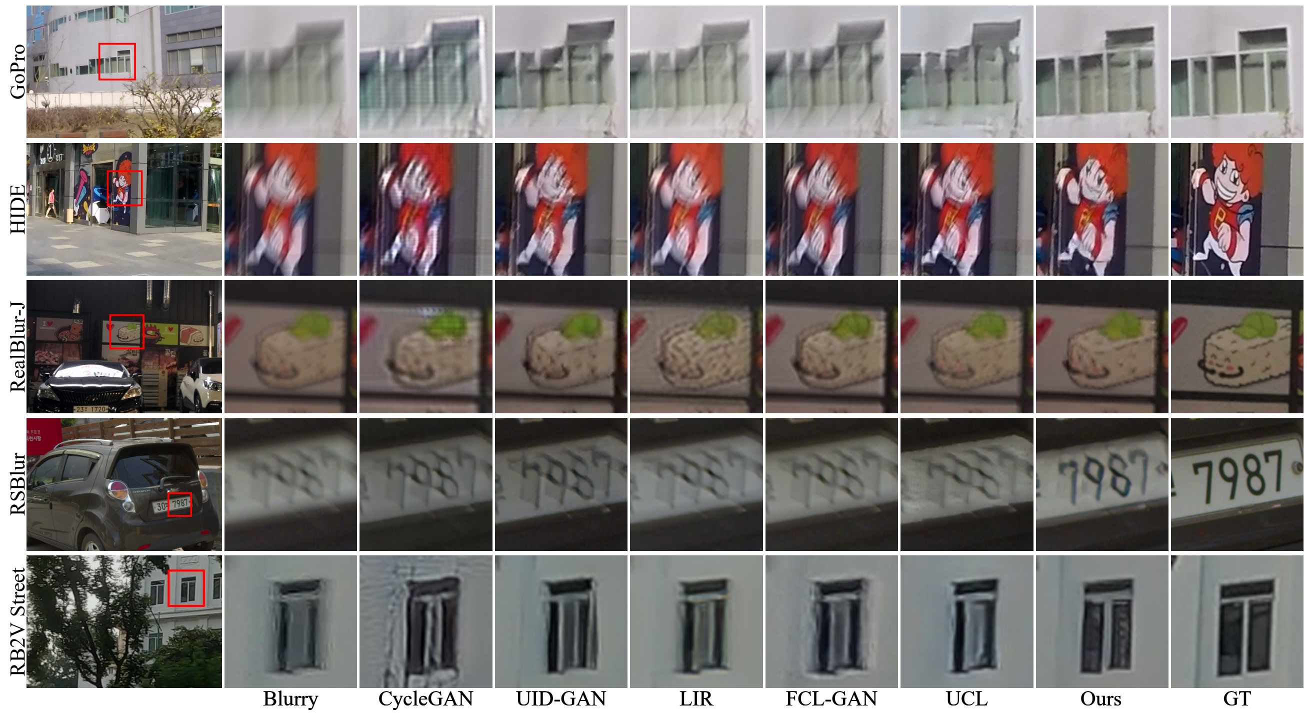

Figure 4. Visual results on GoPro [31], HIDE [42], Realblur-J [38], RSBlur [39], and RB2V_Street [34] datasets. The method is shown at the bottom of each case. Zoom in to see better visualization.

Methods	Components						Metrics	
	Diffusion	TPE	TTformer	Multi-Scale	Joint-Train	WaveLoss	PSNR	SSIM
w/o DM	✗	✓	✓	✓	✗	✓	26.46	0.867
w/o TPE	✓	✗	✓	✓	✓	✓	27.36	0.886
w/o TTformer	✓	✓	✗	✓	✓	✓	27.19	0.884
w/o MS	✓	✓	✓	✗	✓	✓	27.89	0.896
w/o JT	✓	✓	✓	✓	✗	✓	27.65	0.896
w/o WaveLoss	✓	✓	✓	✓	✓	✗	28.01	0.899
Full model	✓	✓	✓	✓	✓	✓	**28.13**	**0.903**

Table 3. Ablation study of each component on the GoPro [31] dataset.

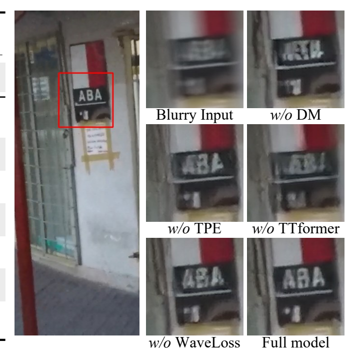

Figure 5. Visualization of ablation.

paper or reproduce results by officially released codes.

Quantitative Comparison. The performance comparisons on the GoPro [31], HIDE [42], RealBlur-R [38], and RealBlur-J [38] test sets are shown in Tab. 1. TP-Diff performs favorably against the latest unpaired training methods (*e.g*., UCL [49]). This is because our texture prior generated by DM can effectively handle spatially varying blurs. Notably, although the latest SEMGUD [3] proposes a self-enhancement strategy that obtains favorable performance, this approach lacks fairness by introducing pre-trained fully supervised models to guide model training. Therefore, we train another version of our model which is optimized with a similar strategy named TP-Diff-*se* for fair comparisons. Tab. 1 shows that our method also achieves better results while using fewer parameters proving its effectiveness.

In addition, we also verify the generalization capabilities of TP-Diff on both realistic RB2V_Street [34] and RSBlur [39] datasets. Tab. 2 shows that TP-Diff outperforms other unpaired training-based methods by 0.66 dB and 0.23 dB on RB2V_Street and RSBlur, respectively. The results verify that TP-Diff has strong generalization capabilities and can effectively handle various realistic blurs through the generated texture prior. Please refer to the supplementary for more quantitative results and efficiency analysis.

Qualitative Comparison. We evaluate the visual quality of different methods in Fig. 4. The results show that TP-Diff has a great improvement in visual quality, especially for areas with detailed textures. For example, in the first and fifth rows of Fig. 4, TP-Diff can recover clearer windows. The results verify that TP-Diff can exploit DM to generate

Exp.	Methods	Sampling	N	PSNR	SSIM
(a)	Latent encoder [5, 25]	Random	-	27.74	0.895
(b)	Enhancement part	Random	256	27.73	0.895
(c)	Transfer part	Random	256	27.88	0.897
(d)	TPE (RGB Space)	Random	256	26.95	0.869
(e)	TPE (Ours)	Cluster	256	28.18	0.902
(f)		Singular	256	27.94	0.898
(g)		Random	128	27.99	0.899
(h)		Random	256	28.13	0.903
(i)		Random	512	28.19	0.904

Table 4. Ablation study of Texture Prior Encoder (TPE).

Method	PSNR	SSIM
w/o DM	26.46	0.867
Memory Bank[52]	27.89	0.894
Sparse Coding[10]	26.66	0.871
Vanilla VQ[48]	27.82	0.895
DM (Ours)	**28.13**	**0.903**

Table 5. Results of different prior generation methods.

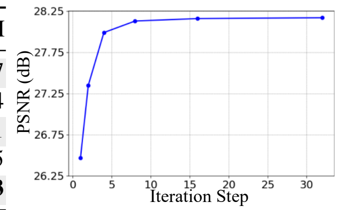

Figure 6. Ablation study of the number of iteration steps T.

Methods	(a)	(b)	(c)	(d)	Ours
PSNR	27.81	27.93	28.04	28.02	**28.13**
SSIM	0.895	0.898	0.901	0.900	**0.903**

Table 6. Results comparisons of different filtering methods.

texture priors that assist in producing finer results. Please refer to the supplementary for more visual results.

4.4. Ablation Study

In this section, we explore the effectiveness of each key component. All ablation studies are trained in the same settings and evaluated on GoPro [31] for fair comparisons.

Effectiveness of Individual Components. We construct ablations in Tab. 3 and Fig. 5 to demonstrate the effectiveness of each component in TP-Diff. We remove each component from the full model to compare performance and keep the same model size for fairness. When DM is not used ("*w/o* DM"), the PSNR is reduced by 1.67dB, which indicates that DM can generate valuable texture prior to enhance deblurring performance. When we remove the TPE ("*w/o* TPE") and TTformer ("*w/o* TTformer"), the PSNR decreases by 0.77dB and 0.94dB, respectively, and the visual quality is severely degraded. It proves that TPE can produce accurate texture prior, and TTformer can effectively remove blurs by exploiting the texture prior. Performance also degrades when we remove the multi-scale learning ("*w/o* MS), only training the DM in the second stage ("*w/o* JT"), and remove the wavelet-based adversarial loss ("*w/o* WaveLoss"). It demonstrates that these components are effective in improving model capability. Overall, our full model achieves a PSNR of 28.13 dB with better visual quality, proving the effectiveness of each component.

Effectiveness of TPE. We constructed several experiments in Tab. 4 to demonstrate the effectiveness of TPEs. (a) and (h) show that the TPE is more conducive to extracting texture prior from unpaired inputs than the latent encoder used in [5, 25]. (b), (c), and (h) show that combining the enhancement and transfer parts results in higher performance, verifying the necessity of each component in TPE. Moreover, (d) and (h) demonstrate that the proposed feature space prior is more robust to feature learning and more stable for training. To demonstrate that the TPE is robust enough to benefit from different unpaired sharp images, we compare the performance of (e) sampling images guided by clustering, (f) reusing the single sharp image (taking the average across ten), and (h) sampling random sharp inputs. Owing to the memory mechanism, the TPE is able to learn beneficial textures even in random sampling. In addition, according to (g)-(i), a larger memory size N contributes to the learning of more diverse textures, but reduces model efficiency. We set N as 256 in our model after trade-offs.

Effectiveness of DM. To demonstrate the necessity of DM in generating texture prior, we compare it with other methods [10, 48, 52] in Tab. 5. The DM performs favorably against the other approaches, proving the advantage of generating texture prior in unpaired restoration. In addition, in Fig. 6, the performance is positively correlated with the iteration steps T and the gain gradually decreases when T is over 8. It shows that only a few iterations are needed to reconstruct the texture prior. We set T to 8 after trade-offs.

Effectiveness of FM-MSA. We compare (a) without filtering, (b) vanilla filtering, (c) deformable filtering [7], and (d) separable filtering [44] in Tab. 6 to show the reliability of proposed adaptive filtering within FM-MSA in Fig. 2(c). They either cannot adaptively change the weights of different regions or are not conducive to capturing non-local blur. Our method allows for handling more complex blurs and performs favorably against other approaches.

5. Conclusion

In this paper, we propose a novel diffusion model-based framework for learning texture priors (TP-Diff). In particular, we recover deblurred images by performing a diffusion model to generate the spatially varying texture prior. To achieve this, we introduce the TPE to encode texture prior of the blurry input, and the TTformer layer to recover texture details using the texture prior. Such a design fully exploits the texture knowledge in unpaired sharp images and provides inspiration for other unpaired restoration tasks. Extensive experiments demonstrate that our TP-Diff outperforms existing SOTA methods.

References

[1] Liang Chen, Faming Fang, Tingting Wang, and Guixu Zhang. Blind image deblurring with local maximum gradient prior. In *CVPR*, pages 1742–1750, 2019. 1

[2] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *ECCV*, pages 17–33. Springer, 2022. 1, 2, 6

[3] Lufei Chen, Xiangpeng Tian, Shuhua Xiong, Yinjie Lei, and Chao Ren. Unsupervised blind image deblurring based on self-enhancement. In *CVPR*, pages 25691–25700, 2024. 1, 2, 5, 6, 7

[4] Xiang Chen, Jinshan Pan, Kui Jiang, Yufeng Li, Yufeng Huang, Caihua Kong, Longgang Dai, and Zhentao Fan. Unpaired deep image deraining using dual contrastive learning. In *CVPR*, pages 2017–2026, 2022. 6

[5] Zheng Chen, Yulun Zhang, Ding Liu, Jinjin Gu, Linghe Kong, Xin Yuan, et al. Hierarchical integration diffusion model for realistic image deblurring. *NeurIPS*, 36, 2024. 2, 3, 5, 6, 8

[6] 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 *ICCV*, pages 4641–4650, 2021. 2

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 8

[8] Jiangxin Dong, Stefan Roth, and Bernt Schiele. Learning spatially-variant map models for non-blind image deblurring. In *CVPR*, pages 4886–4895, 2021. 1, 2

[9] Wenchao Du, Hu Chen, and Hongyu Yang. Learning invariant representation for unsupervised image restoration. In *CVPR*, pages 14483–14492, 2020. 1, 2, 6

[10] Yuchen Fan, Jiahui Yu, Yiqun Mei, Yulun Zhang, Yun Fu, Ding Liu, and Thomas S Huang. Neural sparse representation for image restoration. *NeurIPS*, 33:15394–15404, 2020. 8

[11] Zhenxuan Fang, Fangfang Wu, Weisheng Dong, Xin Li, Jinjian Wu, and Guangming Shi. Self-supervised non-uniform kernel estimation with flow-based motion prior for blind image deblurring. In *CVPR*, pages 18105–18114, 2023. 1, 2, 6

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 27, 2014. 2

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2, 5

[14] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeurIPS*, 35:8633–8646, 2022. 2

[15] Runhua Jiang and Yahong Han. Uncertainty-aware variate decomposition for self-supervised blind image deblurring. In *ACM MM*, pages 252–260, 2023. 1

[16] Neel Joshi, C Lawrence Zitnick, Richard Szeliski, and David J Kriegman. Image deblurring and denoising using color priors. In *CVPR*, pages 1550–1557, 2009. 1

[17] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *NeurIPS*, 35: 23593–23606, 2022. 2

[18] Insoo Kim, Hana Lee, Hyong-Euk Lee, and Jinwoo Shin. Controllable blur data augmentation using 3d-aware motion estimation. In *ICLR*. 2

[19] Insoo Kim, Jae Seok Choi, Geonseok Seo, Kinam Kwon, Jinwoo Shin, and Hyong-Euk Lee. Real-world efficient blind motion deblurring via blur pixel discretization. In *CVPR*, pages 25879–25888, 2024. 2

[20] J Kim. U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *ICLR*, 2020. 6

[21] Diederik P Kingma. Auto-encoding variational bayes. In *ICLR*, 2014. 5

[22] Lingshun Kong, Jiangxin Dong, Jianjun Ge, Mingqiang Li, and Jinshan Pan. Efficient frequency domain-based transformers for high-quality image deblurring. In *CVPR*, pages 5886–5895, 2023. 1, 2, 4

[23] Vladimir Kulikov, Shahar Yadin, Matan Kleiner, and Tomer Michaeli. Sinddm: A single image denoising diffusion model. In *ICML*, pages 17920–17930. PMLR, 2023. 2

[24] Dasong Li, Yi Zhang, Ka Chun Cheung, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. Learning degradation representations for image deblurring. In *ECCV*, pages 736–753. Springer, 2022. 2, 6

[25] Guangyuan Li, Chen Rao, Juncheng Mo, Zhanjie Zhang, Wei Xing, and Lei Zhao. Rethinking diffusion model for multi-contrast mri super-resolution. In *CVPR*, pages 11365–11374, 2024. 2, 3, 5, 8

[26] Chengxu Liu, Xuan Wang, Xiangyu Xu, Ruhao Tian, Shuai Li, Xueming Qian, and Ming-Hsuan Yang. Motion-adaptive separable collaborative filters for blind motion deblurring. In *CVPR*, pages 25595–25605, 2024. 1, 2

[27] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *NeurIPS*, 30, 2017. 1, 2, 6

[28] Boyu Lu, Jun-Cheng Chen, and Rama Chellappa. Unsupervised domain-specific deblurring via disentangled representations. In *CVPR*, pages 10225–10234, 2019. 1, 2, 6

[29] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11461–11471, 2022. 2

[30] Xintian Mao, Yiming Liu, Fengze Liu, Qingli Li, Wei Shen, and Yan Wang. Intriguing findings of frequency selection for image deblurring. In *AAAI*, pages 1905–1913, 2023. 2

[31] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891, 2017. 6, 7, 8

[32] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *CVPR*, pages 2901–2908, 2014. 1

[33] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *CVPR*, pages 1628–1636, 2016. 1

[34] Bang Dang Pham, Phong Tran, A. Tran, C. Pham, R. Nguyen, and Minh Hoai. Hypercut: Video sequence from a single blurry image using unsupervised ordering. In *CVPR*, pages 9843–9852, 2023. 6, 7

[35] Bang-Dang Pham, Phong Tran, Anh Tran, Cuong Pham, Rang Nguyen, and Minh Hoai. Blur2blur: Blur conversion for unsupervised image deblurring on unknown domains. In *CVPR*, pages 2804–2813, 2024. 1, 2, 6

[36] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *CVPR*, pages 3341–3350, 2020. 1, 2

[37] Mengwei Ren, Mauricio Delbracio, Hossein Talebi, Guido Gerig, and Peyman Milanfar. Multiscale structure guided diffusion for image deblurring. In *ICCV*, pages 10721–10733, 2023. 2, 6

[38] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, pages 184–201. Springer, 2020. 1, 6, 7

[39] Jaesung Rim, Geonung Kim, Jungeon Kim, Junyong Lee, Seungyong Lee, and Sunghyun Cho. Realistic blur synthesis for learning image deblurring. In *ECCV*, pages 487–503. Springer, 2022. 6, 7

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 6

[42] Ziyi Shen, Wenguan Wang, Xiankai Lu, Jianbing Shen, Haibin Ling, Tingfa Xu, and Ling Shao. Human-aware motion deblurring. In *ICCV*, pages 5572–5581, 2019. 6, 7

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2, 5

[44] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *CVPR*, pages 11166–11175, 2019. 8

[45] Xiaole Tang, Xile Zhao, Jun Liu, Jianli Wang, Yuchun Miao, and Tieyong Zeng. Uncertainty-aware unsupervised image deblurring with deep residual prior. In *CVPR*, pages 9883–9892, 2023. 1

[46] Dmitrii Torbunov, Yi Huang, Haiwang Yu, Jin Huang, Shinjae Yoo, Meifeng Lin, Brett Viren, and Yihui Ren. Uvcgan: Unet vision transformer cycle-consistent gan for unpaired image-to-image translation. In *WACV*, pages 702–712, 2023. 6

[47] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, and Chia-Wen Lin. Stripformer: Strip transformer for fast image deblurring. In *ECCV*, pages 146–162. Springer, 2022. 2, 4

[48] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 30, 2017. 8

[49] Yongzhen Wang, Xuefeng Yan, Fu Lee Wang, Haoran Xie, Wenhan Yang, Xiao-Ping Zhang, Jing Qin, and Mingqiang Wei. Ucl-dehaze: Towards real-world image dehazing via unsupervised contrastive learning. *IEEE TIP*, 2024. 6, 7

[50] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *CVPR*, pages 16293–16303, 2022. 2

[51] Jia-Hao Wu, Fu-Jen Tsai, Yan-Tsung Peng, Chung-Chi Tsai, Chia-Wen Lin, and Yen-Yu Lin. Id-blau: Image deblurring by implicit diffusion-based reblurring augmentation. In *CVPR*, pages 25847–25856, 2024. 1, 2

[52] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 8

[53] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. Diffir: Efficient diffusion model for image restoration. In *ICCV*, pages 13095–13105, 2023. 2, 3, 5

[54] Yanyang Yan, Wenqi Ren, Yuanfang Guo, Rui Wang, and Xiaochun Cao. Image deblurring via extreme channels prior. In *CVPR*, pages 4003–4011, 2017. 1

[55] Tian Ye, Sixiang Chen, Wenhao Chai, Zhaohu Xing, Jing Qin, Ge Lin, and Lei Zhu. Learning diffusion texture priors for image restoration. In *CVPR*, pages 2524–2534, 2024. 2

[56] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, pages 2849–2857, 2017. 6

[57] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, pages 14821–14831, 2021. 2

[58] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pages 5728–5739, 2022. 2, 4, 6

[59] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, pages 4791–4800, 2021. 6

[60] Youjian Zhang, Chaoyue Wang, and Dacheng Tao. Neural maximum a posteriori estimation on unpaired data for motion deblurring. *IEEE TPAMI*, 2023. 1, 2

[61] Suiyi Zhao, Zhao Zhang, Richang Hong, Mingliang Xu, Yi Yang, and Meng Wang. Fcl-gan: A lightweight and real-time baseline for unsupervised blind image deblurring. In *ACM MM*, pages 6220–6229, 2022. 1, 2, 5, 6

[62] Zhihang Zhong, Ye Gao, Yinqiang Zheng, and Bo Zheng. Efficient spatio-temporal recurrent neural network for video deblurring. In *ECCV*, pages 191–207. Springer, 2020. 1

[63] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 6