Learning Deblurring Texture Prior from Unpaired Data with Diffusion Model

Supplementary Material

In this supplementary material, Sec. A1 illustrates the detailed architecture in our TP-Diff. Sec. A2 describes the detailed training and inference algorithms. Sec. A3 analyses the model efficiency. Sec. A4 describes in detail the difference between the texture prior in our method and HiDiff [5]. Sec. A5 provides a detailed explanation of the self-enhancement strategy mentioned in the experiments. Sec. A6 analyses the upper bound of the performance. Sec. A7 describes the dataset used in our method. Sec. A8 analyzes the limitations. Finally, Sec. A9 shows more quantitative and qualitative comparison results.

A1. Architecture Details

As described in Sec. 3.1 of the main paper. The deblurring network and reblurring network together form the entire cycle structure designed for removing and synthesizing blur, respectively. Within the deblurring network, to fully leverage the texture prior and enhance the model capacity, we incorporate the Texture Transfer Transformer (TTformer) at multiple scales and feed the texture prior $\hat{z}$ into them.

Specifically, we illustrate the detailed architecture of the deblurring network as shown in Fig. A1. We follow the existing approach [58] to learn features by stacking some TTformer layers on each scale, where the number of layers is marked. In each TTformer layer, a filter-modulated multi-head self-attention (FM-MSA, see Fig. 2(c) of the main paper) and a transform-modulated feed-forward network (TM-FFN, see Fig. 2(d) of the main paper) are included. The parameters of the deblurring network are 11.8M. The reblurring network is based on the standard U-Net structure of residual blocks with a parameter size of 29.2 MB, and it is used only during training.

In addition, we use a neural network consisting of five stacked ResBlocks, denoted as ϵ_θ, to estimate the noise. The purpose of using ResBlocks as the denoising network is to ensure the same resolution of inputs and outputs while minimizing the model parameters. The parameters of the denoising network are 0.1M.

A2. Algorithm

The first and second stage training algorithms for TP-Diff are shown in Alg. 1 and Alg. 2, respectively. The inference algorithm for TP-Diff is shown in Alg. 3.

A3. Efficiency

We report the parameters and runtime compared to other state-of-the-art methods in the main paper, this section analyzes in detail the effectiveness of the core components in our methods. In particular, during inference, the parameters of our TP-Diff are 11.89M and the computational overhead is 52.7G MACs. Notably, our computational overhead is also lower than the latest method SEMGUD (TP-Diff:52.7G *vs.* SEMGUD:63.6G). In our TP-Diff, the diffusion model parameter used for prior reconstruction is 0.12M and the runtime is 5.2ms when inputting 256×256 on 3090 GPU, where a total of 9.2G MACs is consumed for the 8 iterations. Although we use a diffusion model, it only costs a small portion of the overall model overhead, proving the efficiency of our approach.

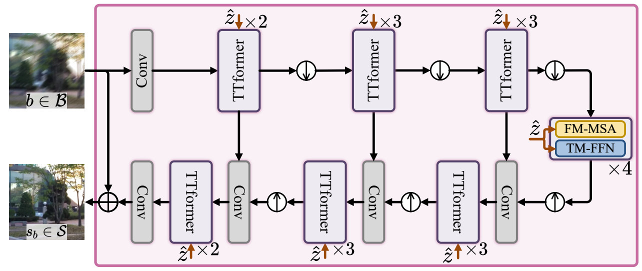

Figure A1. Network structure of deblurring network.

Algorithm 1 TP-Diff Training: Stage One

Input: Texture Prior Encoder (TPE), deblurring network, reblurring network.

Output: Trained TPE, traiend deblurring network, trained reblurring network.

for $s \in \mathcal{S}, b \in \mathcal{B}$ **do**
  $z = \text{TPE}(s, b)$. (paper Eqs. (1)-(4))
  $s_b = \text{DeblurringNetwork}(b, z)$
  $b_s = \text{ReblurringNetwork}(s)$
  $z = \text{TPE}(s_b, b_s)$. (paper Eqs. (1)-(4))
  $\hat{s} = \text{DeblurringNetwork}(b_s, z)$
  $\hat{b} = \text{ReblurringNetwork}(s_b)$
  Calculate $\mathcal{L}_{s1}$ loss (paper Eq. (12)).
end for
Output the trained TPE, traiend deblurring network, trained reblurring network.

A4. Prior Differences Compared to HiDiff [5]

It should be emphasized that our texture prior is quite different from the sharp prior in HiDiff [5], and the reasons are as follows:

- The capability of the obtained prior is different. Our texture prior in different spaces is only used to handle blur-

Algorithm 2 TP-Diff Training: Stage Two

Input: Trained TPE, traiend deblurring network, trained reblurring network, denoising network, $\beta_t (t \in [1, T])$.
Output: Trained denoising network, trained deblurring network.

```
Init: $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_T = \prod_{i=0}^{T} \alpha_i$.
Init: The deblurring network copies the parameters of trained deblurring network.
Init: The reblurring network copies the parameters of trained reblurring network.
Init: The TPE copies the parameters of trained TPE and freezes them.
for $s \in \mathcal{S}, b \in \mathcal{B}$ do
    $z = \mathrm{TPE}(s, b)$. (paper Eqs. (1)-(4))
    Diffusion Process:
    We sample $z_T$ by $q(z_T \mid z) = \mathcal{N}(z_T; \sqrt{\bar{\alpha}_T} z, (1 - \bar{\alpha}_T) \mathbf{I})$ (paper Eq. (13))
    Denoising Process:
    $\hat{z}_T = z_T$
    $c = \mathrm{Conv}(b)$
    for $t = T$ to $1$ do
        $\hat{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}} (\hat{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\hat{z}_t, c, t)) + \sqrt{1-\alpha_t} \epsilon_t$ (paper Eq. (15))
    end for
    $\hat{z} = \hat{z}_0$
    $s_b = \mathrm{DeblurringNetwork}(b, \hat{z})$
    $b_s = \mathrm{ReblurringNetwork}(s)$
    $z = \mathrm{TPE}(s_b, b_s)$. (paper Eqs. (1)-(4))
    Diffusion Process:
    We sample $z_T$ by $q(z_T \mid z) = \mathcal{N}(z_T; \sqrt{\bar{\alpha}_T} z, (1 - \bar{\alpha}_T) \mathbf{I})$ (paper Eq. (13))
    Denoising Process:
    $\hat{z}_T = z_T$
    $c = \mathrm{Conv}(b_s)$
    for $t = T$ to $1$ do
        $\hat{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}} (\hat{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\hat{z}_t, c, t)) + \sqrt{1-\alpha_t} \epsilon_t$ (paper Eq. (15))
    end for
    $\hat{z} = \hat{z}_0$
    $\hat{s} = \mathrm{DeblurringNetwork}(b_s, \hat{z})$
    $\hat{b} = \mathrm{ReblurringNetwork}(s_b)$
    Calculate $\mathcal{L}_{s2}$ loss (paper Eq. (16)).
end for
Output the trained denoising network and trained deblurring network.
```

ring in the corresponding region. The spatial diversity in our texture prior is reflected in that the prior with different regions is only used to handle the corresponding region. In contrast, HiDiff uses a set of out-of-order priors with a specific quantity, it cannot explicitly represent the blurring in different regions. We compare their performance in Tab. 4 of the main paper, demonstrating the advantages of the generated prior in our TP-Diff.

- The application scenarios are different. The supervision used to generate prior in HiDiff comes from paired data and is not feasible for unpaired inputs. Benefiting from our TPE, TP-Diff can learn texture priors from unpaired data and is robust enough for different sharp inputs. Please note that it is the first attempt to introduce the diffusion model to unpaired restoration and could inspire other unpaired tasks.
- The structure of the denoising network used to generate the prior is different. Our TP-Diff uses CNNs to compose the denoising network, while the HiDiff uses the MLPs. In contrast, our denoising network has fewer parameters (TP-Diff: 0.12M *vs.* HiDiff: 0.44M) and comparable runtimes (TP-Diff: 5.2ms *vs.* HiDiff: 3.4ms) when inputting 256×256 on 3090 GPU.

A5. About Self-Enhancement Strategy in SEMGUD [3]

In Tab. 1 of the main paper, 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.

Algorithm 3 TP-Diff Inference

Input: Trained denoising network, trained dehazing network, $\beta_t (t \in [1, T])$, blurry images $b \in \mathcal{B}$.
Output: Deblurred images S_b.

1: Init: $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_T = \prod_{i=0}^{T} \alpha_i$.
2: **Denoising Process**:
3: Sample $z_T \sim \mathcal{N}(0, 1)$
4: $\hat{z}_T = z_T$
5: $c = \mathrm{Conv}(b)$
6: **for** $t = T$ to 1 **do**
7: $\quad \hat{z}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\hat{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\hat{z}_t, c, t)) + \sqrt{1-\alpha_t}\epsilon_t$ (paper Eq. (**15**))
8: **end for**
9: $\hat{z} = \hat{z}_0$
10: $s_b = \mathrm{DeblurringNetwork}(b, \hat{z})$
11: Output deblurred images s_b.

As stated in Sec. D of SEMGUD's supplementary, for training stability, it introduces the pre-trained NAFNet (33.69 dB PSNR on GoPro) as the extra deblurring model before estimating the prior from blurry inputs, thus bringing more performance gains. In contrast, TP-Diff is more fair by training directly from scratch using unpaired data. Therefore, we train another version of our model which is optimized with a similar strategy named TP-Diff-*se* for fair comparisons. The experimental results show that we also obtain better performance when using the same strategy (TP-Diff-*se*:30.16dB *vs.* SEMGUD:29.06dB).

A6. About Upper Bound

It is worth emphasizing that in the first stage (*i.e.*, not involving the diffusion model), our model uses unpaired blurry-sharp images as input. In this case, the model performance is limited by the selection of unpaired sharp images, and the performance reaches an upper bound if fully paired blurry-sharp images are used directly as input. Theoretically, this also represents the upper bound of the second stage can be reached. If paired data inputs are used directly, the model performance reaches an upper bound (GoPro: 33.46dB/0.965, HIDE: 31.52dB/0.945). Moreover, it can also be noted from the results of HiDiff in Tab. 1 of the main paper, our method also generates a more beneficial texture prior when using fully paired inputs and yields better results.

A7. More Dataset Details

We evaluate the our method on widely-used datasets: **GoPro** [31], **HIDE** [42], **RealBlur** [38], **RB2V_Street** [34], and **RSBlur** [39]. **GoPro** [31] dataset includes 2,103 pairs for training and 1,111 pairs for testing. **HIDE** [42] dataset only includes 2,025 images pairs for testing. **RealBlur** [38] dataset contains two subsets: **RealBlur-R** and **RealBlur-J**. Each subset contains 980 pairs for testing. **RB2V_Street** [34] dataset includes 9,000 pairs for training and 2,053 pairs for testing. **RSBlur** [39] dataset includes 8,878 pairs for training and 3,360 pairs for testing.

During training, our method requires unpaired blurry image sets $\mathcal{B}$ and sharp image sets $\mathcal{S}$. For fair comparisons, we follow existing works [3, 15, 35] to construct training data. Specifically, we split the training set of GoPro (containing 2,103 image pairs), RSBlur (containing 13,358 image pairs), and RB2V_Street (containing 11,000 image pairs) dataset into two disjoint subsets that capture different scenes with a specific ratio of 0.6:0.4. In the first subset, we select blurry images to form the blurry image set $\mathcal{B}$, while in the second subset, we choose sharp images to construct the sharp set $\mathcal{S}$. The statistics of training image sets and test image sets are reported in Tab. A1.

Based on this, 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.

A8. Limitation

Although our texture prior can handle spatially varying blur, the resolution of the texture prior that needs to be generated increases as the input resolution increases. This means that the computational effort of the diffusion model will increase. Therefore, it is expected to make the diffusion model learn a set with a fixed number of texture priors to learn sharp features so as to avoid increasing computational costs significantly.

In addition, a more powerful reblurring is one of the important factors in improving performance. However, the core of TP-Diff enables a powerful DM to assist the deblurring process by predicting the unknown texture prior. To realize this, we propose TPE to supervise DM training and learn to generate spatially varying texture priors. Future we will further explore the DM for reblurring performance.

A9. More Results

In this section, we first provide experiments to verify the effectiveness of the diffusion model. We then analyze the sensitivity of the hyper-parameters in the loss function. Finally, we show more visualization results.

Effect of Hyper-parameter λ_{Wave}. To explore the impact of the wavelet-based adversarial loss we presented in Eq. (11), we discuss the different λ_{Wave} as shown in Fig. A2. The experiment results show that too small λ_{Wave} cannot effectively preserve the texture structure,

Datasets	Number of data samples		
	Train-$\mathcal{B}$	Train-$\mathcal{S}$	Test Pairs
GoPro [31]	1,262	841	1,111
HIDE [42]	-	-	2,025
RealBlur-R [38]	-	-	980
RealBlur-J [38]	-	-	980
RB2V_Street [34]	5,400	3,600	2,053
RSBlur [39]	8,115	5,410	3,361

Table A1. Statistics of datasets used in our method.

while too large λ_{Wave} affects the illumination of the image and reduces the performance. Therefore, we empirically set λ_{Wave} to 0.2 in our model.

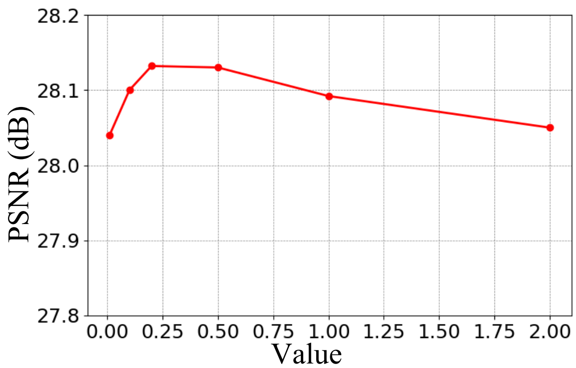

Figure A2. Sensitivity analysis of λ_{Wave}.

Effect of Hyper-parameter K. To show the reliability of adaptive filtering within FM-MSA in Fig. 2(c) of the main paper, we analyze the effect of kernel size K on describing complex blurs for adaptive filtering in Fig. A3. The performance positively correlates with K. It demonstrates the powerful potential of our adaptive filtering to handle complex blurs. Although a larger K will allow more pixels to be referenced, it will also increase the computational overhead. We finally set K to 5.

Experiments of Cross-Validation. In Tab. A2, we follow [18, 19] using RealBlur-J and RSBlur for cross-validation to verify the generalization ability. Results show that our TP-Diff is able to achieve better generalization ability compared to other unpaired training methods. It is worth noting that it is unfair to compare the cross-validation results of our method with other generalized deblurring methods, since the unpaired inputs are already inherently more challenging than the paired inputs. In addition, the core of TP-Diff is to assist the deblurring process by introducing a

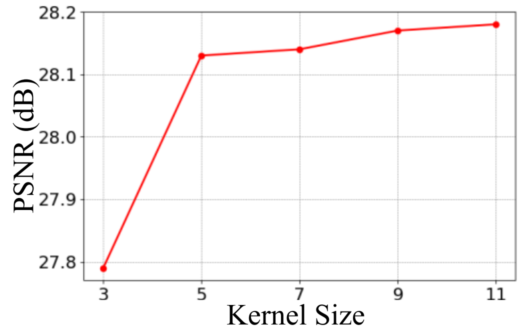

Figure A3. Effect of the number of kernel size K.

Methods	UVCGANv2 [46]	UCL [49]	TP-Diff
PSNR	24.85	24.56	25.45
SSIM	0.682	0.701	0.735

Table A2. Results of cross-validated experiments.

diffusion model that predicts beneficial texture prior, rather than learning the blurry degradation template.

More Visual Results To further verify the effectiveness of our method, we show more comparison results among the proposed TP-Diff and other advanced methods on six different benchmarks. The results on **GoPro** [31], **HIDE** [42], **RealBlur-J** [38], **RealBlur-R** [38], **RSBlur** [39], and **RB2V_Street** [34] are shown in Fig. A4, Fig. A5, Fig. A6, Fig. A7, Fig. A8, and Fig. A9, respectively.

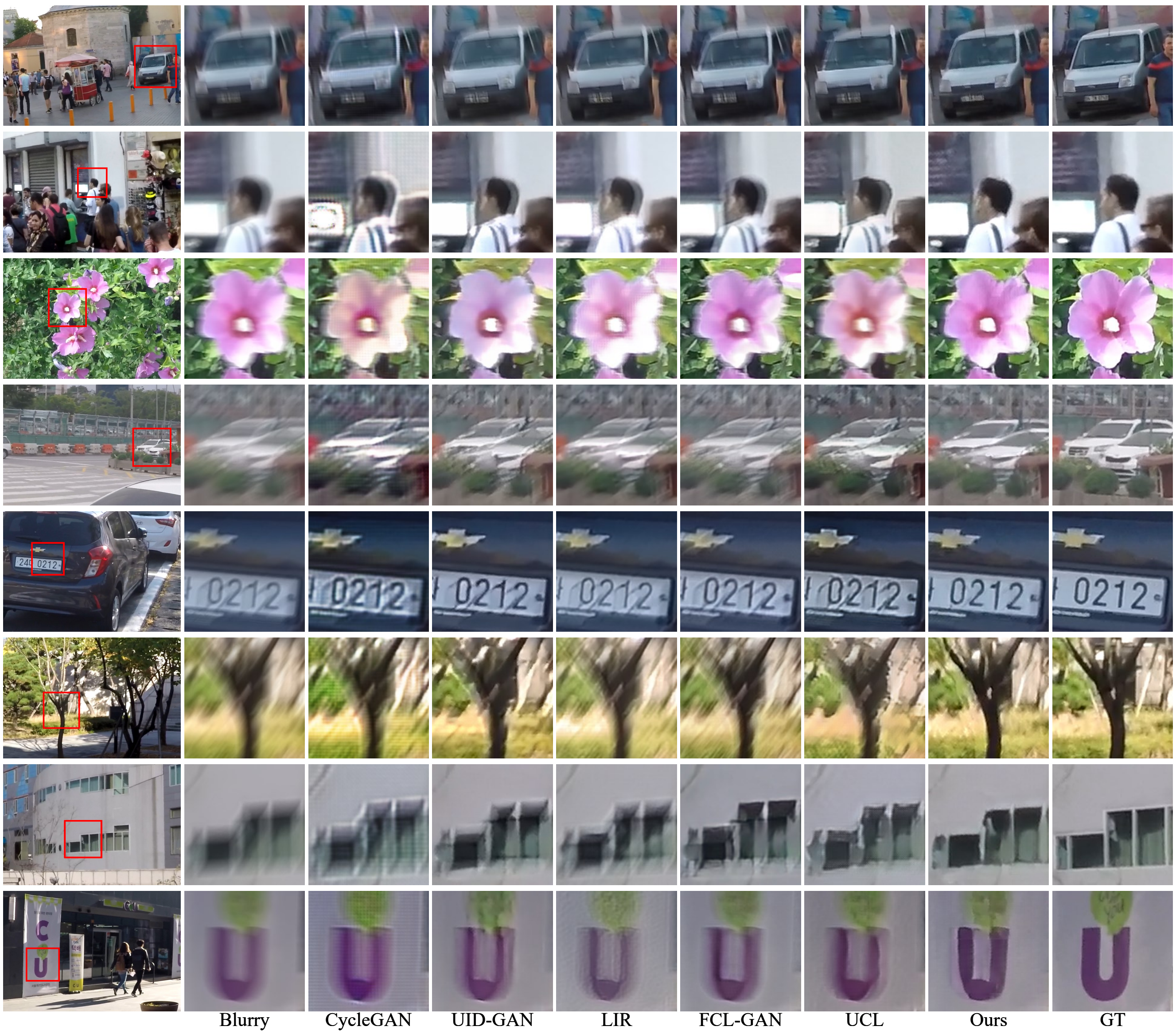

Figure A4. Visual results on GoPro [31] dataset. The method is shown at the bottom of each case. Zoom in to see better visualization.

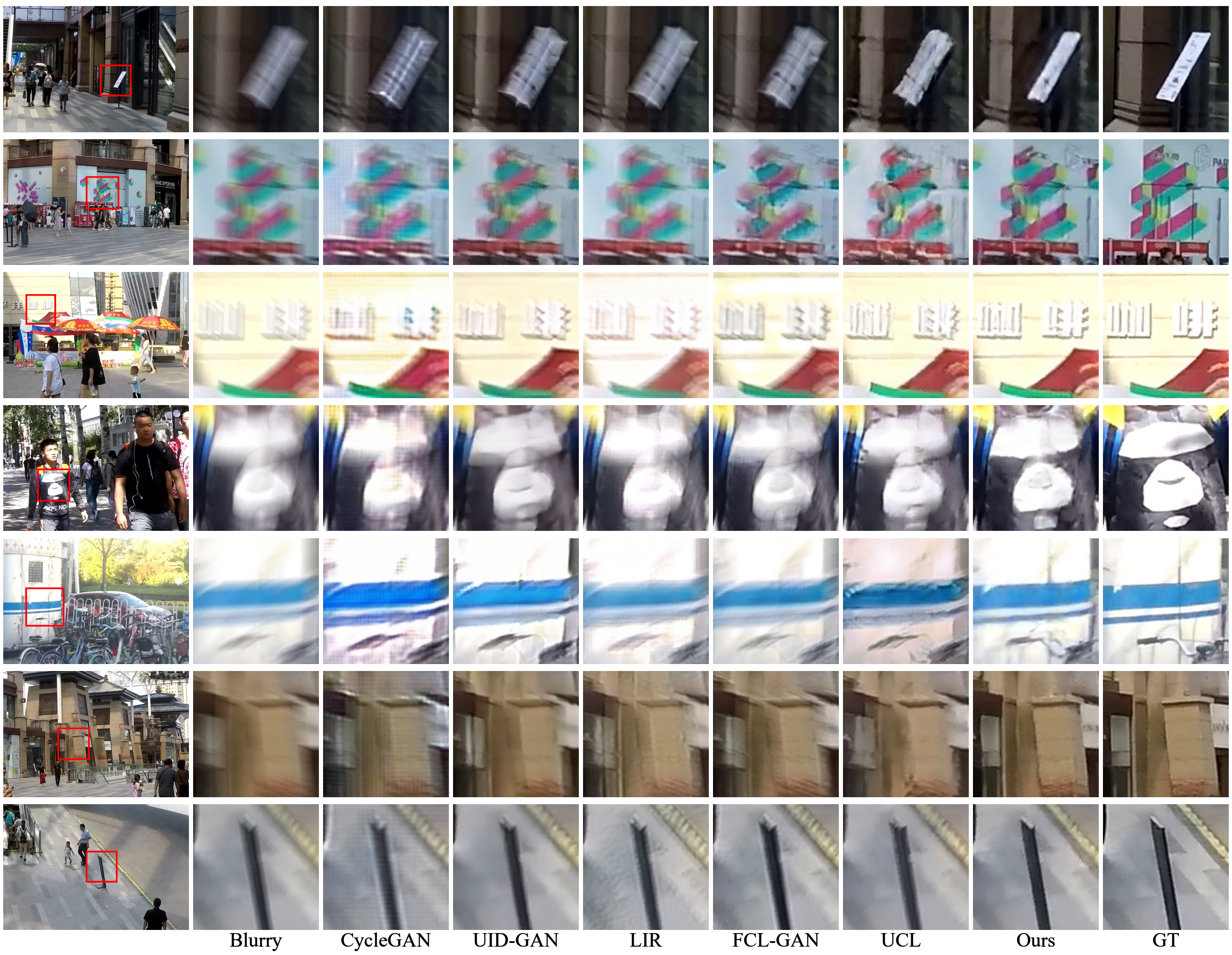

Figure A5. Visual results on HIDE [42] dataset. The method is shown at the bottom of each case. Zoom in to see better visualization.

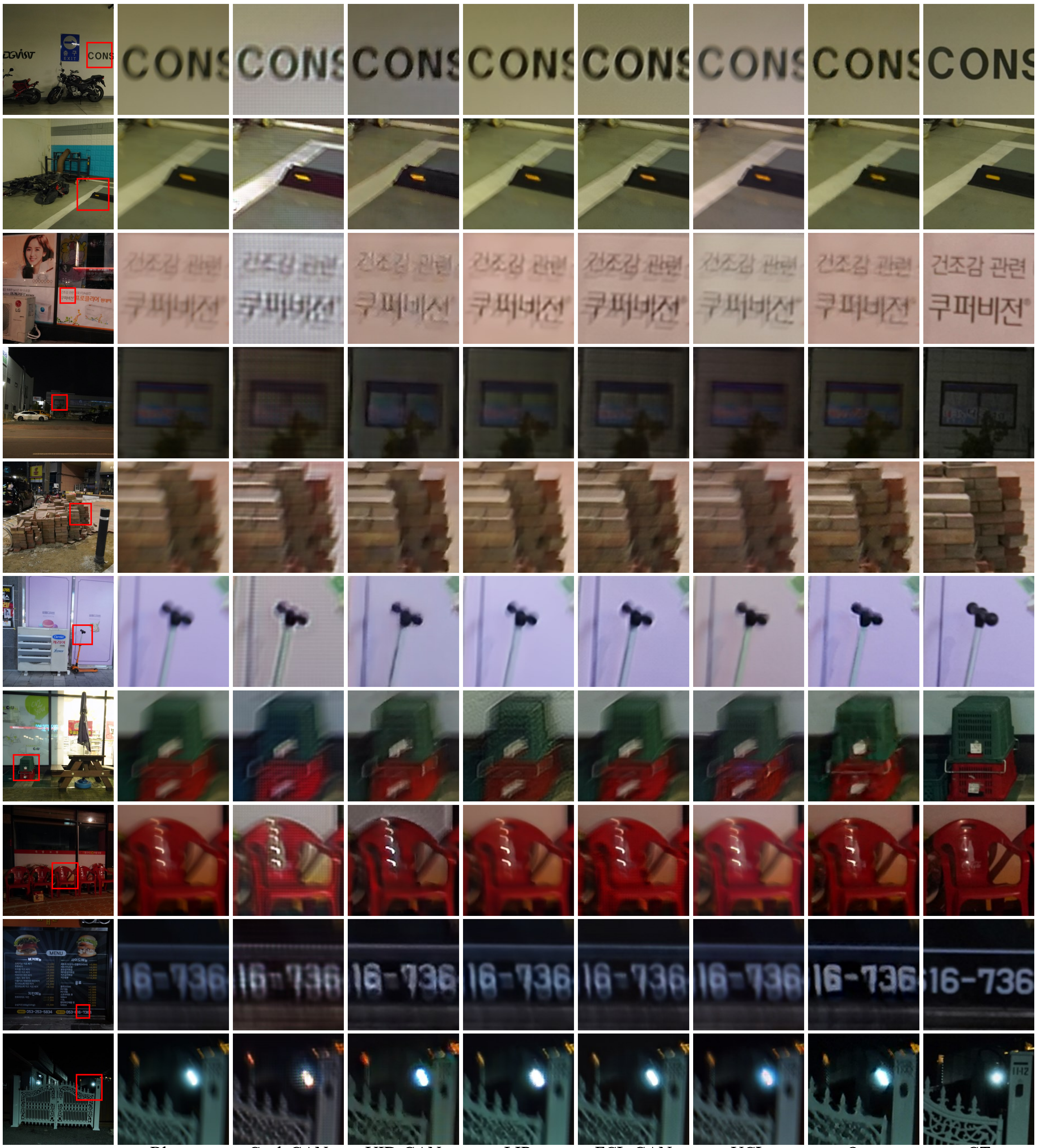

Figure A6. Visual results on RealBlur-J [38] dataset. The method is shown at the bottom of each case. Zoom in to see better visualization.

Figure A7. Visual results on RealBlur-R [38] dataset. The method is shown at the bottom of each case. Zoom in to see better visualization.

Figure A8. Visual results on RSBlur [39] dataset. The method is shown at the bottom of each case. Zoom in to see better visualization.

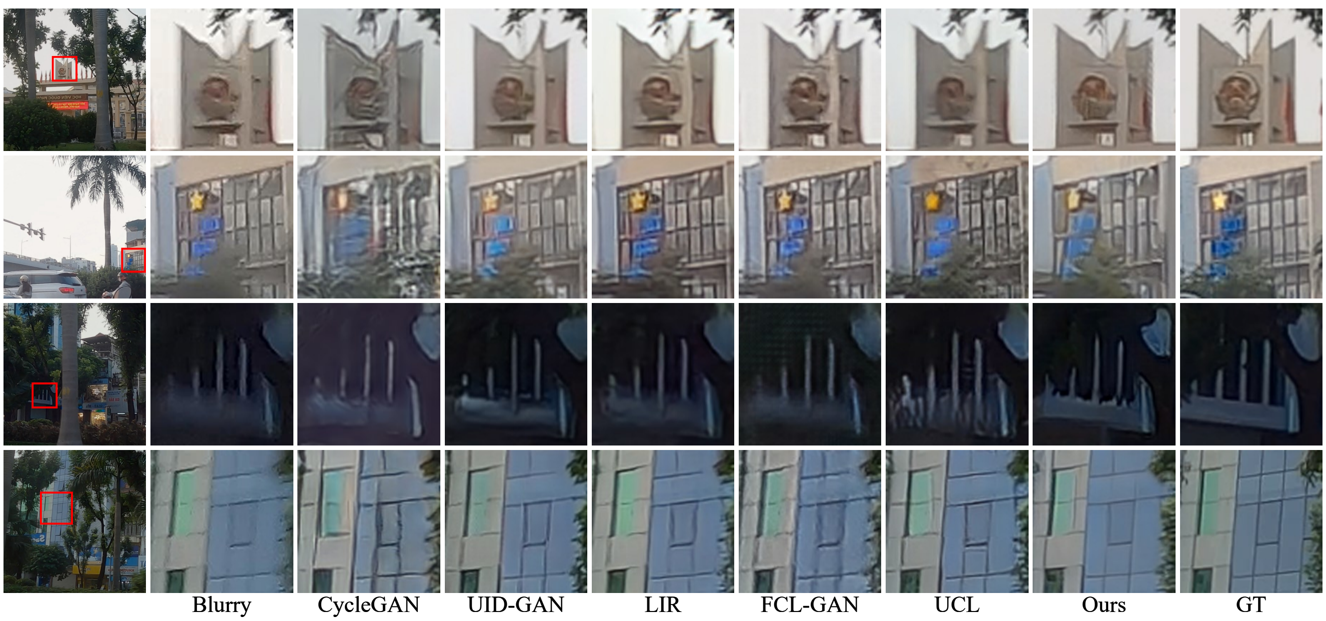

Figure A9. Visual results on RB2V_Street [34] dataset. The method is shown at the bottom of each case. Zoom in to see better visualization.

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, 3

[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, 1

[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, 3

[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, 4

[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, 4

[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, 3, 4, 5

[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, 3, 4, 10

[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, 3

[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, 3, 4, 8

[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, 3, 4, 9

[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, 3, 4

[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, 4

[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, 4

[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, 1

[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