Supplementary Material
Context-aware Pretraining for Efficient Blind Image Decomposition

Chao Wang[1,2*], Zhedong Zheng[3], Ruijie Quan[1], Yifan Sun[2], Yi Yang[1†]
[1]ReLER, CCAI, Zhejiang University
[2]Baidu Inc.
[3]Sea-NExT Joint Lab, School of Computing, National University of Singapore

1. Implementation Details

Appearance Flow Sampling: As described in Section **3.2**, we adopt the similar Gaussian sampling strategy [13] to avoid the bad local minima problem. The sampling process can be written as follow:

$$\mathbf{F}_{out} = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\alpha_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_{i,j}} \mathbf{F}_{i,j}, \tag{7}$$

where n is the kernel size. $\mathbf{F}_{i,j}$ represents the features around the sample center and $\mathbf{F}_{out}$ is the output feature. $\alpha_{i,j}$ is the sampling weight calculated as:

$$a_{i,j} = \exp\left(-\frac{\Delta h^2 + \Delta v^2}{2\sigma^2}\right), \tag{8}$$

where Δh and Δv are the horizontal and vertical distance between the sampling center and feature $\mathbf{F}_{i,j}$, respectively. σ is the variance of the sampling kernel.

Noise Construction: During the pre-processing stage, we randomly add different combinations of degradation to the clean images. For a fair comparison, we use the same setting for raindrop construction as BIDeN [5]. The model for rainstreak and snow is:

$$I_{rain/snow}(x) = J(x)(1 - m(x)) + A * m(x), \tag{9}$$

and the model for haze is:

$$I_{haze}(x) = J(x)t(x) + A * (1 - t(x)), \tag{10}$$

where x is the image pixel, I is the observed intensity, J is the scene radiance, A is the global atmospheric light, and m is the mask of rain streak and snow. t denotes the transmission map.

The model for reflection is formulated as:

$$I_{reflection}(x) = T(x) + R(x) * V(x), \tag{11}$$

and the model for watermark composition is:

$$I_{watermark}(x) = J(x)(1 - w(x)) + A * w(x), \tag{12}$$

where T is the transmission layer, R is the reflection layer, and V denotes the vignette mask. J is the scene radiance, A is the global atmospheric light, and w represents the watermark image.

As for shadow removal, we use the same setting on SRD [12] dataset with the masks generated by DHAN [3]. The reflection image R is processed by a Gaussian smoothing kernel with a random kernel size, where the size is in the range of 3 to 17 pixels during training, and fixed to 11 pixels during testing. For both Eqs. (9), (10) and (12), we set A between [0.8, 1.0] during training, and fix A = 0.9 for testing. Before the random combination, all masks are randomly rotated with angles [0°, 90°, 180°, 270°] for more robust pretraining.

Dataset & **Experimental Settings:** We compared the proposed CPNet mainly with Restormer [20], All-in-one [7], BIDeN [5], MPRNet [21], RCDNet [15], DHAN [3] and Auto-Exposure [4] method. The performance of the compared methods is acquired through former publicly available pretrained models or implementation codes [1] [2] [3] [4] [5] [6]. Note that since the original code of All-in-one [7] is not publicly available, the qualitative results are based on our implementation. For a fair comparison, we trained and tested all methods under the same setting as BIDeN [5] with the same mask dataset[7].

In particular, Task-I is based on the CityScape [2] dataset, where we use the original test set as our training set (2975), and the validation set as our test set (500). The test set for all source components contains a fixed number of 500 images. The mask dataset contains four different masks with various intensities, including rainstreak (1620), rain-

∗ Work done during an internship at Baidu.
† Corresponding author: Yi Yang.

[1] https://github.com/leftthomas/Restormer
[2] https://github.com/JunlinHan/BID
[3] https://github.com/swz30/MPRNet
[4] https://github.com/hongwang01/RCDNet
[5] https://github.com/vinthony/ghost-free-shadow-removal
[6] https://github.com/tsingqguo/exposure-fusion-shadow-removal
[7] https://drive.google.com/drive/folders/1wUUKTiRAGVvelarhsjmZZ_1iBdBaM6Ka

Table 6. Quantitative results of several structural variants on raindrop + snow removal on CityScape [2] dataset. The best performances under each case are marked in **bold**. (S)W-MSA represents the two successive Swin-Transformer blocks [10].

Module	Vanilla Encoder	BIDeN	MHA		FFN		Ours
			(S)W-MSA	MDTA	LeFF	GDFN	MDTA+GDFN
PSNR ↑	28.71	29.16	32.25	31.77	31.15	31.89	**32.02**
SSIM ↑	0.822	0.881	0.918	0.887	0.864	0.903	**0.910**
FLOPs ↓	325G	344G	216G	90G	67G	93G	**102G**

drop (3500), haze (2975), and snow (3500). The masks for rainstreak are acquired from Rain100L and Rain100H [17]. For raindrop masks, we model the droplet shape and property using the meta-ball model [1], following the same setting as BIDeN [5]. Haze masks are acquired from Foggy CityScape [14] with three different intensities, while the masks for snow are selected from Snow100K [9].

As for Task-II, the training set contains 3661 natural images as ground truth, where 861 images are adopted from the training set of [11], 1800 images are borrowed from the training set of Rain1800 [17], and the rest 1000 images are selected from the training set of Snow100K [9]. The training masks are identical to Task-I. After training under three cases, these models are respectively tested on three task-specific real-world datasets, which consist of 1000 images from SPAData [16] test set for rainstreak, 249 images from DeRaindrop [11] for raindrop, and 1329 images from DeSnowNet [9] for snow removal.

As for Task-III, we select the shadow dataset from [5]. The dataset is based on SRD [12], which consists of 2680 paired shadow masks, shadow-free images, and shadow images as the training set, and the test set contains 408 images for each type of degradation. The shadow masks are generated with DHAN [3] method. The algorithm for adding reflection to images is acquired from [22], we select 3120 images from the reflection subset as the reflection layer. We use 3000 paired RGB watermark images and masks for the watermark effect, which are acquired from the training set of LVM [8]. Following the data split of SRD, the training set contains 2580 reflection layer images and 2460 watermark images/masks.

2. Network Details

Transformer Encoder: As mentioned in Section 3.1, this paper mainly focuses on exploring a context-aware pretraining scheme, while the transformer design in CPNet is rather flexible. We replace the Multi-Head Attention (MHA) and Feed Forward Networks (FFN) with several cutting-edge modules, which are Shifted Window based Self-Attention (SW-MSA) [10], Multi-Dconv head Transposed Attention (MDTA) [20], Locally-Enhanced Feed-Forward (LeFF) [19] and Gated Dconv Feed-Forward Network (GDFN) [20]. As shown in Table 6, each variant is evaluated in terms of Peak Signal-to-Noise Ratio (PSNR), structural similarity (SSIM), and floating point of operations (FLOPs). All experiments are conducted under the same setting and environment for joint raindrop and snow removal tasks. It can be clearly observed that more sophisticated modules can further boost the performance of our CPNet. In this work, we adopt a more lightweight structure [20] (MDTA + GDFN) considering the balance between accuracy and efficiency.

Fine-tuning Network: The brief structure of our texture finetuning network is shown in Figure 10, which is a simple autoencoder structure. During the pretraining stage, we adopt a ℓ_1 loss for structure reconstruction encouraging the model to be more robust under different types of degradation. As for finetuning, we use a ℓ_2 reconstruction loss to impose further supervised constraint on image details such as texture, which is deployed as:

$$\mathcal{L}^t_{\ell_2} = \|I_{gen} - I_{gt}\|_2 . \tag{13}$$

where I_{gen} is the predicted images and I_{gt} means the ground truth. Meanwhile, we adopt a default perceptual loss [6] with a pretrained VGG16, which consists of two parts, *i.e.*, the content loss $\mathcal{L}^t_c$ and style loss $\mathcal{L}^t_s$:

$$\mathcal{L}^t_c = \sum_{i=1}^{N} \frac{1}{HWC} \left| \phi^{gt}_{pool_i} - \phi^{gen}_{pool_i} \right|_1 , \tag{14}$$

$$\mathcal{L}^t_s = \sum_{i=1}^{N} \frac{1}{C*C} \left| \frac{1}{HWC} \left(\phi^{style_{gt}}_{pool_i} - \phi^{style_{gen}}_{pool_i} \right) \right|_1 , \tag{15}$$

where ϕ_{pool_i} represents the feature map after $pool_i$ layer, $\phi^{style}_{pool_i} = \phi_{pool_i} \phi^T_{pool_i}$, and N is the number of feature maps. The overall training loss for the refinement network N_r is $\mathcal{L}^t_{\ell_2} + \lambda_1(\mathcal{L}^t_c + \mathcal{L}^t_s)$, λ_1 denotes the hyper-parameter which are set as 0.01 in this paper.

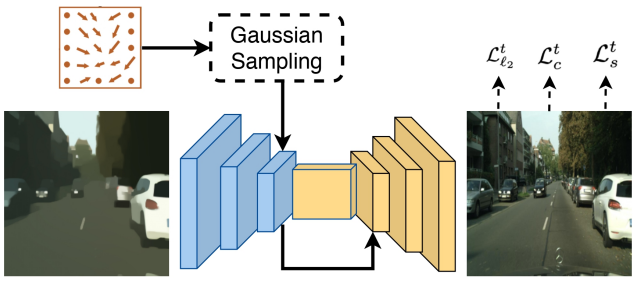

Figure 10. Architecture of the texture refinement network.

Table 7. Quantitative comparisons on different fine-tuning strategies. N_r is the texture refinement network. MHP represents the multi-head prediction module.

Finetuning	Input	MPRNet	N_r	N_r + MHP
NIQE ↓	4.87	4.10	4.13	**4.02**
BRISQUE ↓	27.82	28.66	25.58	**25.11**

Table 8. Detailed architectures of the proposed modules, in which H_s means the appearance head, H_t is the structure head, D_{img} is the discriminator judging whether the image is true or false, and D_{att} represents the discriminative branch predicting the attribute label of degradation type with dimension n. Conv(c,k,s) denotes a standard convolutional layer with channel c, kernel size k and stride s. GConv() represents the gated convolution layer [18] and σ is the sigmoid activation.

H_t	H_s	D_{img}	D_{att}
GConv(256,3,1),BN,GELU	GConv(256,3,1),BN,GELU	Conv(64,3,2),IN,RELU	
× 3	× 3	Conv(128,3,2),IN,RELU	
with skip connection	with skip connection	Conv(256,3,2),IN,RELU	Conv(256,3,2),IN,RELU
GConv(256,3,1),BN,GELU	GConv(256,3,1),BN,GELU	Conv(512,3,2),IN,RELU	Conv(512,3,2),IN,RELU
GConv(128,3,1),BN,GELU	GConv(128,3,1),BN,GELU	Conv(1024,3,2),IN,RELU	Conv(1024,3,2),IN,RELU
	GConv(64,3,1),BN,GELU	FC(1024),IN,RELU	FC(1024),IN,RELU
	GConv(32,3,1),BN,tanh	FC(1)	FC(n), σ

As shown in Table 7, we further unlock the Multi-Head Prediction (MHP) module along with refinement network N_r for fine-tuning, leading to a higher performance since the gradients from the textual information can directly guide the training of the appearance flow map. We also observe a marginal performance improvement with a more complex refinement network, which further indicates the effectiveness of our pre-trained model. We also show the detailed architecture of the multi-head prediction module as well as the multi-head discriminator in Table 8.

3. More Results

Figure 11 and Figure 12 show more qualitative comparisons and results for Task-I. More results on real-world Task-II are given in Figure 13, Figure 14 and Figure 15. Figure 16 and Figure 17 show the results for Task-III under seven different noise combinations. It can be seen that our method stably produces well-structured results with finer details while remaining robust to different noise combinations and training strategies.

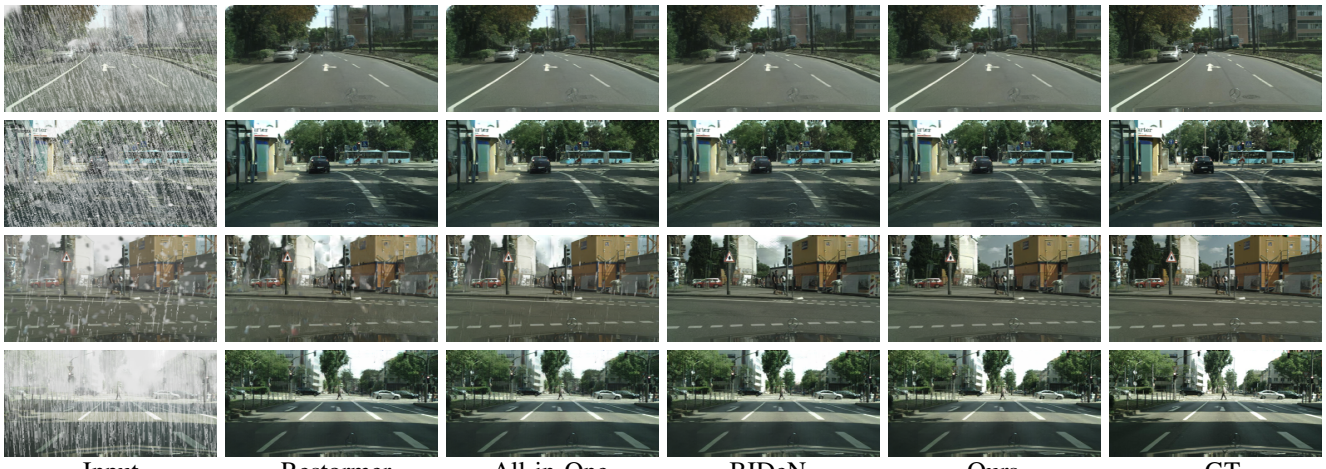

Figure 11. More comparisons with other SOTA methods on Task-I. Please zoom in to see the details.

Figure 12. More qualitative results of the proposed CPNet under each case in Task-I. Case (1): rain streak, (2): rain streak + snow, (3): rain streak + light haze, (4): rain streak + heavy haze, (5): rain streak + moderate haze + raindrop, (6) rain streak + snow + moderate haze + raindrop. Every two rows represent one case. Please zoom in to see the details.

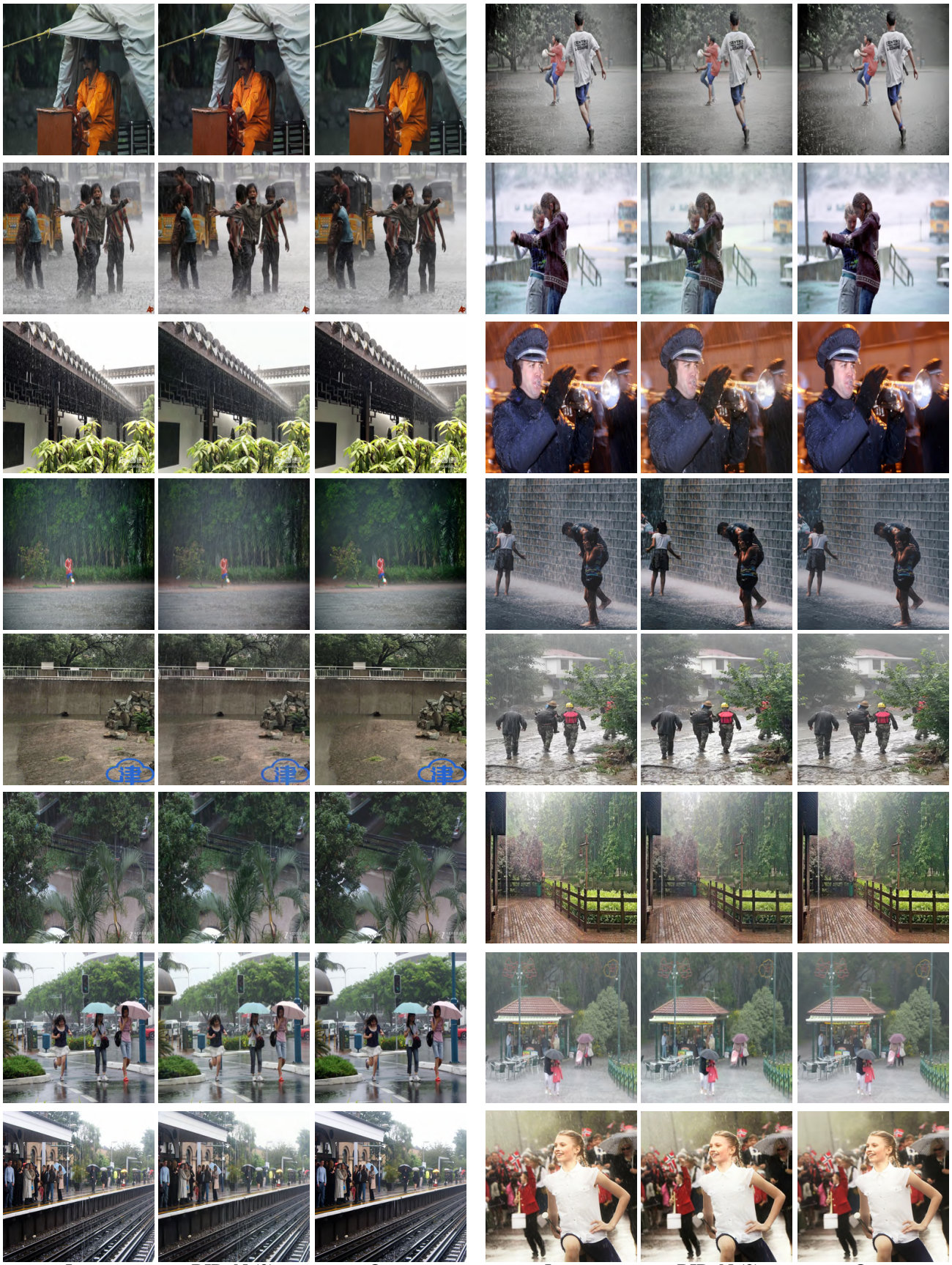

Figure 13. More qualitative comparisons on real-world rainstreak removal scenario. BIDeN(3) represents the model that is jointly trained with three noise combinations: rainstreak + raindrop + snow. Please zoom in to see the details.

Figure 14. More qualitative comparisons on real-world raindrop removal scenario. Please zoom in to see the details.

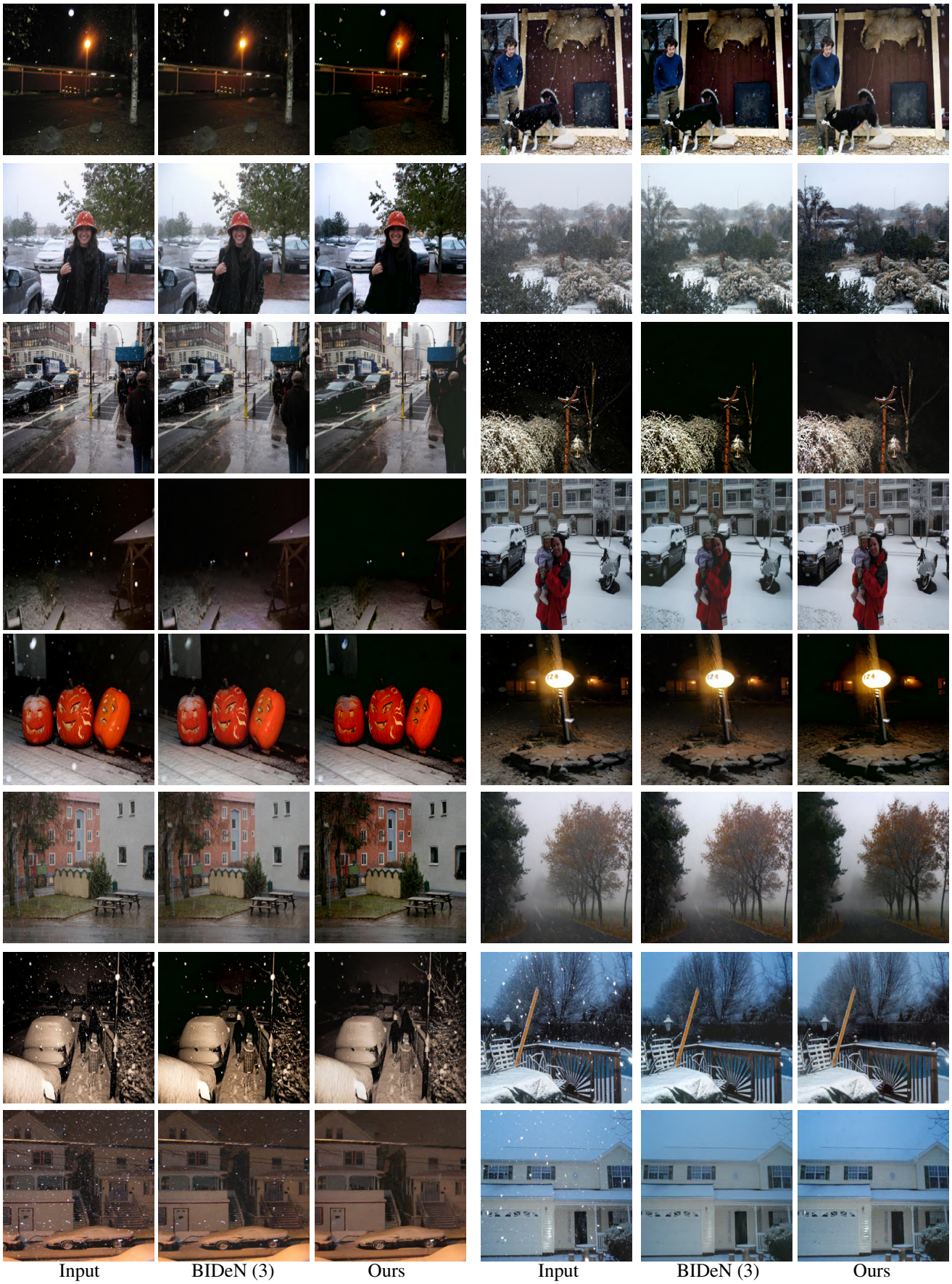

Figure 15. More qualitative comparisons on real-world snow removal scenario. Please zoom in to see the details.

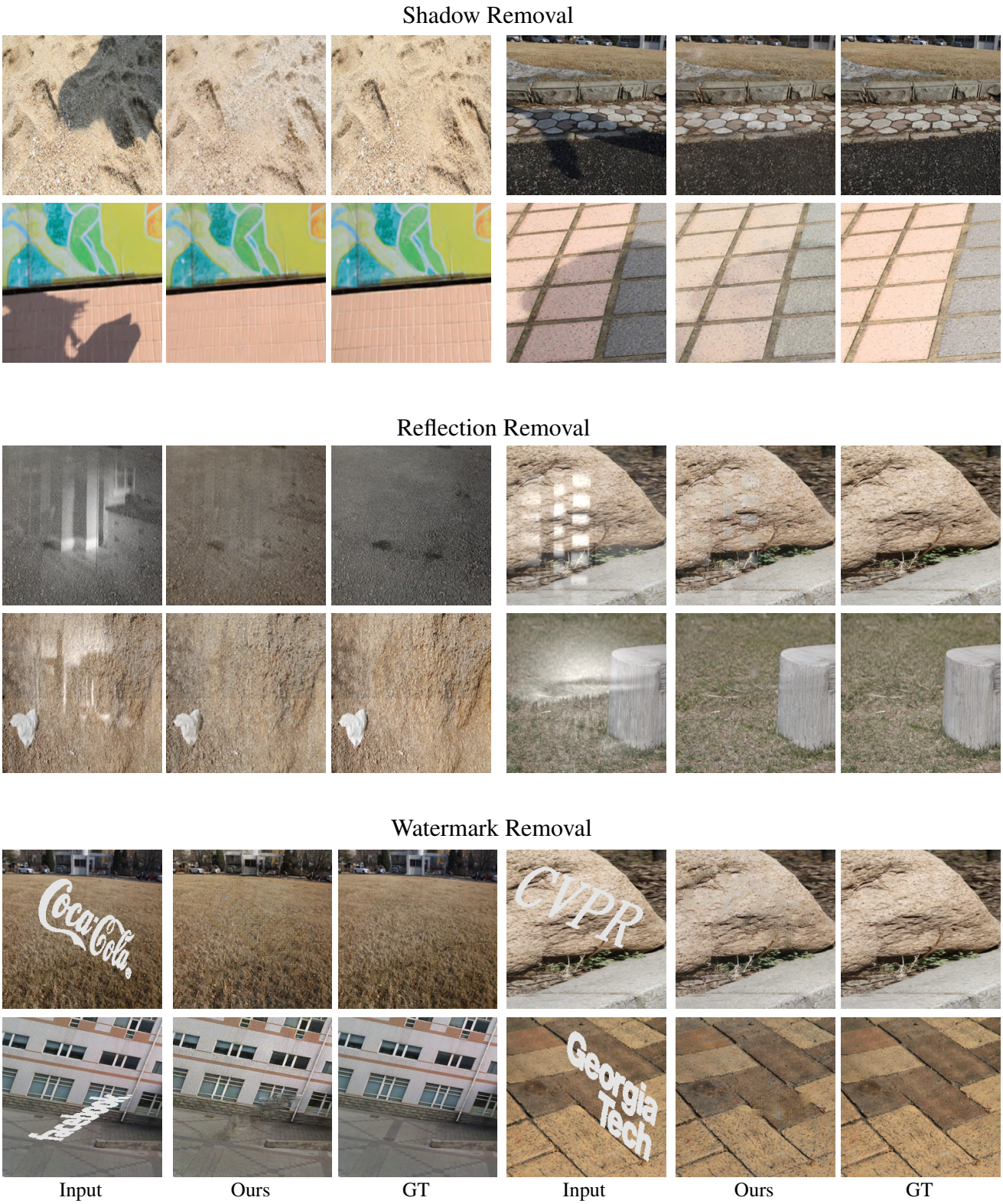

Figure 16. More qualitative results on Task-III. Please zoom in to see the details.

Figure 17. More results on Task-III with different noise combinations. Please zoom in to see the details.

References

[1] J. F. Blinn. A generalization of algebraic surface drawing. *TOG*, 1(3):235–256, 1982.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[3] X. Cun, C. M. Pun, and C. Shi. Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting gan. In *AAAI*, 2020.

[4] L. Fu, C. Zhou, Q. Guo, F. Juefei-Xu, H. Yu, W. Feng, Y. Liu, and S. Wang. Auto-exposure fusion for single-image shadow removal. In *CVPR*, 2021.

[5] J. Han, W. Li, P. Fang, C. Sun, J. Hong, M. A. Armin, L. Petersson, and H. Li. Blind image decomposition. In *ECCV*, 2022.

[6] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[7] R. Li, R. T. Tan, and L. F. Cheong. All in one bad weather removal using architectural search. In *CVPR*, 2020.

[8] Y. Liu, Z. Zhu, and X. Bai. Wdnet: Watermark-decomposition network for visible watermark removal. In *WACV*, 2021.

[9] Y. F. Liu, D. W. Jaw, S. C. Huang, and J. N. Hwang. Desnownet: Context-aware deep network for snow removal. *TIP*, 27(6):3064–3073, 2018.

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[11] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. In *CVPR*, 2018.

[12] L. Qu, J. Tian, S. He, Y. Tang, and R. W. Lau. Deshadownet: A multi-context embedding deep network for shadow removal. In *CVPR*, 2017.

[13] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li. Structureflow: Image inpainting via structure-aware appearance flow. In *ICCV*, 2019.

[14] C. Sakaridis, D. Dai, and L. Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 126(9):973–992, 2018.

[15] H. Wang, Q. Xie, Q. Zhao, and D. Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, 2020.

[16] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, 2019.

[17] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *CVPR*, 2017.

[18] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019.

[19] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu. Incorporating convolution designs into visual transformers. In *ICCV*, 2021.

[20] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M. H. Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

[21] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M. H. Yang, and L. Shao. Multi-stage progressive image restoration. In *CVPR*, 2021.

[22] X. Zhang, R. Ng, and Q. Chen. Single image reflection separation with perceptual losses. In *CVPR*, 2018.