

Channel Consistency Prior and Self-Reconstruction Strategy Based Unsupervised Image Deraining

Supplementary Material

In this paper, we propose a novel Channel Consistency Prior and Self-Reconstruction Strategy based Unsupervised Image Deraining framework, **CSUD**, to address the lack of paired data and the poor generalization in real-world scenarios. Extensive experiments on multiple synthetic and real-world datasets verify that our method achieves excellent deraining performance. What’s more, as shown in Fig. 1, our CSUD achieves the most promising performance on the real-world captured images compared to both supervised and unsupervised deraining methods, which demonstrates strong generalization capability of our CSUD. This supplementary material mainly includes the following contents:

- More detailed explanations of the proposed channel consistency prior and self-reconstruction strategy;
- The specific structure of certain used networks;
- More implementation details of the experiments mentioned in the main document;
- Additional experiment results and analysis;
- Discussion and limitations of our method.

1. More Explanations of Channel Consistency Prior

In the main document, we have provided a detailed introduction to the channel consistency prior (CCP) of rain streaks: in RGB rainy images, most rain streaks tend to be consistent across the R, G, and B channels, and the cycle subtractions of R, G, and B channels in rainy images will almost contain no rain streaks and should be close to the cycle subtractions in clean images. To present the conclusions derived from CCP more clearly and intuitively, we provide a more detailed visualization of the CCP in rainy images along with the corresponding clean images in Fig. 2, including synthetic rain, real-world rain, and real-world nighttime rain. We can clearly observe that while there are significant differences in the background parts among the three channels of rainy images, the rain streaks tend to be consistent, and the cycle subtractions of R, G, and B channels in rainy images tend to be consistent with that of their corresponding clean images.

Currently, the rain in an image is typically modeled using the linear superposition model: $Y = X + R$, where Y is the rainy image, R is the rain layer, and X is the image background. Most image deraining researches [2–4, 7, 12, 18, 24] focuses on the removal of the rain layer R in images. One of the reasons why many unsupervised image deraining models perform poorly is that, while they remove rain layer, they struggle to maintain the integrity of other background information in the image. In this work, we also mainly focus on

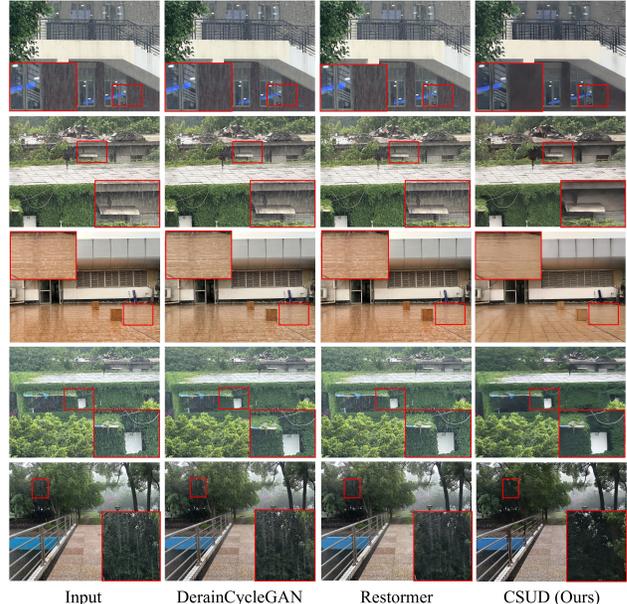


Figure 1. Deraining results on the real rainy images captured by ourselves in real-world scenarios. Compared with the supervised method Restormer [22] and the unsupervised method DerainCycleGAN [18], our CSUD exhibits extremely strong generalization capability and achieves the best visual results.

removing rain layer R from rainy images. During training, the derainer learns the mapping between pseudo-rainy images and real clean images X . If the pseudo-rainy images maintain the main background information of X , differing from clean images only by the presence of the rain streaks, the derainer will learn the optimal mapping relationship. Thus, we introduce the Channel Consistency Loss (CCLoss) to constrain the generator G to produce pseudo-rainy images that retain most of the image content consistent with the original clean images X aside from the rain streaks. Through the constrain of CCLoss, the derainer can better learn the desired mapping.

In addition, we separately analyze the special cases which include nighttime artificial light sources. As shown in Fig. 3, we provide a visualization of CCP **under the nighttime artificial light scenario**. While the rain streaks in rainy image appear yellowish due to interference from artificial light, the rain streaks in the residual image do not exhibit such yellowish tones, and most rain streaks still conform to CCP among R,G,B channels. According to the linear superposition model, rainy images can be understood as the superposition of the rain layer and the clean background.

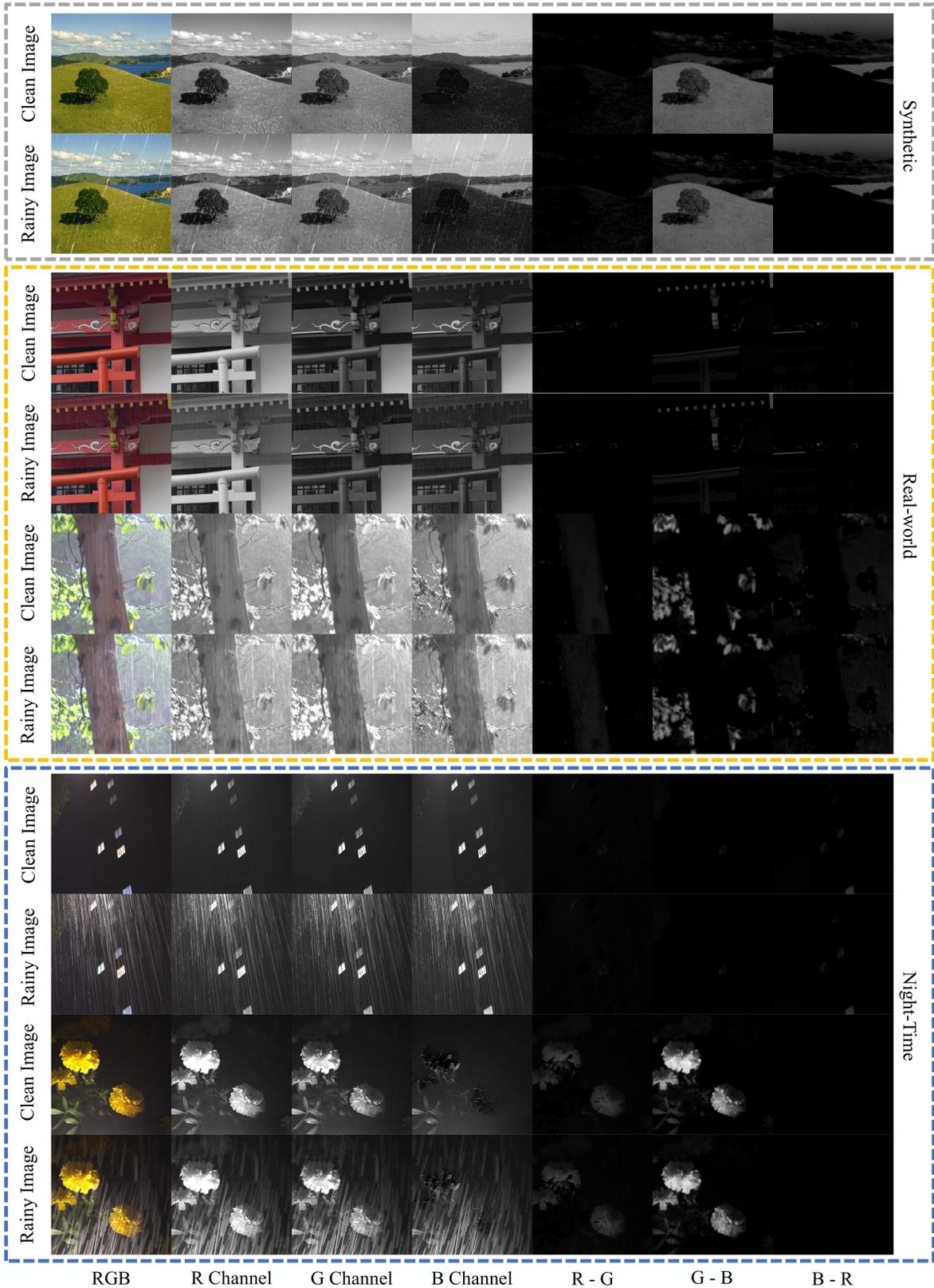


Figure 2. Visualization of channel consistency prior in rainy images. From top to bottom, the 3 sets of images are synthetic, real-world, and real-world nighttime images, respectively. The first column presents the clean and rainy RGB images; the second, third and fourth columns present their R, G, B channels, respectively; the fifth, sixth, and seventh columns present the cycle subtractions of R, G, and B channels of clean and rainy images, respectively.

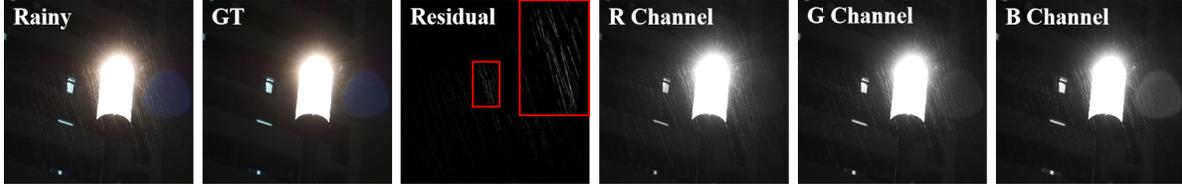


Figure 3. Visualization of channel consistency prior under the nighttime artificial light scenario.

Therefore, the yellowish hue in rain streaks is the color of the background introduced by artificial light. Furthermore, in Fig.7, Tab.1, and Tab.2 of the main text, we test our model on real-world datasets RealRain1K-L and RealRain1K-H which **include numerous nighttime artificial light sources**, and our CSUD achieves remarkable results.

Certainly, there may be some rain streaks that do not adhere to the CCP, but this does not significantly affect the network’s performance. This is because, besides CCP, the SR strategy and overall unsupervised framework with the three additional constraints based on CycleGAN [27] can also ensure the accurate and effective generation of pseudo-paired rainy images, as well as the deraining and generalization performance of CSUD. The CCLoss serves as a further auxiliary enhancement to the overall framework, aiming to transfer rain streaks while preserving more background details as much as possible, which makes our framework more robust. It is the combined effort of our unsupervised framework, CCLoss, and the SR strategy that achieves the outstanding unsupervised deraining performance of CSUD.

2. More Explanation of Self-Reconstruction Strategy

As described in main document, x_{der} restored by derainer Der is not adopted in SRLoss for generator L_{SR-G} . However, x_{der} can also be used in L_{SR-G} to constrain the training process of generator G , but we do not add it in the final implementation in order to make the SR loss for generator L_{SR-G} and SR loss for derainer L_{SR-Der} on the same scale and reduce the calculation of the training process, which has little effect on the performance of the network. Additionally, as shown in Fig. 4, we present more visualizations of the effects of SR strategy on the performance of generator G . With SR strategy, the generator effectively alleviates the redundant information transfer problem, ensuring that higher-quality pseudo rainy images are generated.

3. Detailed Network Structures.

As described in the main document, the proposed CSUD mainly consists of a derainer, a generator, and a discriminator. To balance the performance and computational complexity, we adopt the simple CNN-based image restoration baseline NAFNet [1] (the version of width32) as the default derainer.

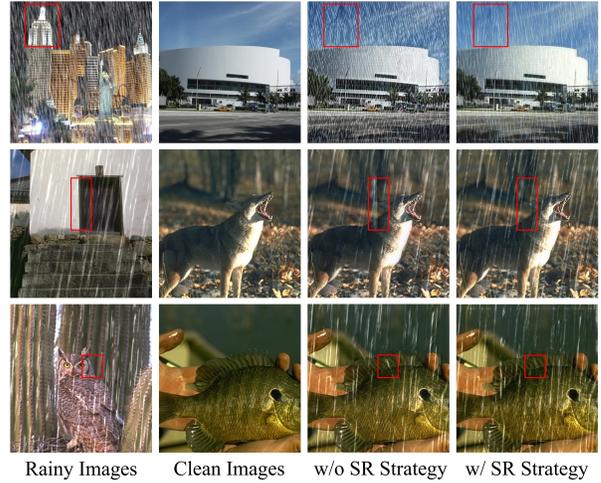


Figure 4. The effects of SR strategy on the performance of generator. The first and second columns present the input rainy and clean images of the generator, respectively; the third column presents pseudo rainy images generated by the generator without SR strategy; the fourth column presents pseudo rainy images generated by the generator with SR strategy.

Architecture of the Generator. The architecture of the generator used in our network is shown in Fig. 5, it consists of a clean feature extraction module (CFEM), a rain information extraction module (RIEM) and 6 residual blocks. The RIEM is based on a U-Net architecture, comprising a downsampling layer followed by an upsampling layer, while the CFEM simply utilize a convolutional layer. Specifically, the two convolutional layers in RIEM (denoted as "conv") have input channels, output channels, kernel size, stride, and padding settings of [3, 64, 7, 1, 3] and [64, 64, 7, 1, 3], respectively. The downsampling layer in RIEM is a convolutional layer with input channels, output channels, kernel size, stride, and padding set to [64, 128, 4, 2, 1]. The upsampling layer in RIEM uses a 'bilinear' interpolation, followed by a convolutional layer with input channels, output channels, kernel size, stride, and padding set to [128, 64, 3, 1, 1]. The convolutional layer in CFEM has input channels, output channels, kernel size, stride, and padding set to [3, 64, 7, 1, 3]. Each residual block consists of two 3×3 convolution layers with ReLU activation function. The generator learns the rain characteristics of rainy images to guide the synthesis

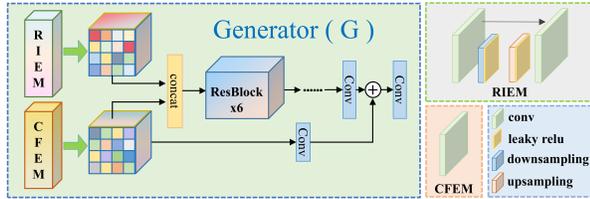


Figure 5. Detailed network structures of the generator.

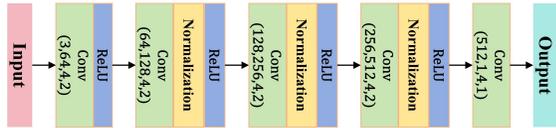


Figure 6. Detailed network structures of the discriminator.

of clean images towards rainy ones, providing ample pseudo rainy images paired with the clean ones for the derainer.

Architecture of the Discriminator. In our network, we use a Patch-GAN [8] discriminator, as shown in Fig. 6. The discriminator is starting with a 4×4 convolution layer with ReLU activation function, followed by three intermediate layers, each of which consists of instance normalization between the convolution layer and the activation function, and ending with a 4×4 convolution layer with a stride of 1.

4. More Explanation of Experiment Setting

As described in the main document, following [10, 18], we conduct experiments from two aspects: (1) unsupervised deraining performance and (2) generalization performance. In experiment (2), we only train our model on the synthetic dataset Rain100L [19], and then test on various real-world and nighttime datasets. This experimental setup is designed to better demonstrate the strong cross-domain generalization ability of our CSUD framework, specifically its deraining performance when faced with various rain streak distributions, rather than suggesting that our CSUD should be trained exclusively on synthetic datasets for optimal performance. Our CSUD can also be trained on unpaired real-world datasets and achieve better performance. In the main document, we have conducted experiments on the real-world RealRain-1k-L and RealRain-1k-H datasets according to the experiment (1) settings. Trained on unpaired real-world datasets, CSUD achieves better results on the two real-world datasets, even surpassing some classic supervised methods.

5. Experiment Details

Datasets. Detailed descriptions of the datasets employed are provided in Tab. 1. In experiment (1), we use the corresponding different training sets to train independent models for Rain100L [19], Rain100H [19], Rain800 [9], RealRain1K-L [10], and RealRain1K-H [10] test sets respectively. Notably, we utilize the model trained on Rain100L to test on

Table 1. Detailed description of the datasets utilized.

Datasets	Rain100L [19]	Rain100H [19]	Rain12 [11]	Rain800 [9]	RealRain1K-L [10]
Train	200	200	0	700	784
Test	100	100	12	100	224
Rain Type	Synthetic	Synthetic	Synthetic	Synthetic	Real-world

Datasets	RealRain1K-H [10]	SPA-data [16]	RainDS [15]	Internet-Data [17]	Night-Rain [23]
Train	784	638,492	150	0	5000
Test	224	1000	98	147	500
Rain Type	Real-world	Real-world	Real-world	Real-world	Night-Time

Rain12 [11] dataset. As for experiment (2), we only train our model on Rain100L [19], and then test on the 6 real-world and night-time test sets, including RealRain1K-L [10], RealRain1K-H [10], SPA-data [16], RainDS [15], Internet-Data [17], and Night-Rain [23]. It is worth noting that RainDS includes multiple subsets, including synthetic and real subsets, with the two subsets further divided into rain streaks, rain drops, and a mixture of rain streaks and drops. Since our method focuses on removing rain streaks and experiment (2) is to evaluate generalization performance on real-world and night-time test sets, so we only select the rain streaks subset from the real RainDS subset for testing. All other comparison models are also tested on this subset.

Implementation Details. Our framework is implemented by PyTorch [13] with a GeForce RTX 3090 GPU. For training, we adopt the Adam optimizer [5] ($\beta_1 = 0.9$, $\beta_2 = 0.999$) to train our network. We train the framework for 200 epochs with the initial learning rate of $1e^{-4}$, followed by another 100 epochs with a learning rate of $1e^{-5}$. All training images are randomly cropped to 256×256 patches in an unpaired learning manner, and the batch size is set to 2. The hyperparameters of SSIM loss (λ_1), perceptual loss (λ_2), and SRLoss for derainer (λ_3) are set to 1, 0.2 and 0.5 respectively, while CCLoss (α_1) and SRLoss for generator (α_2) are set to 10 and 5 respectively. Notably, we add perceptual loss to our framework is not to improve perceptual quality of our results, but to enhance the stability of the unsupervised training process. Because we find that only using L1 loss as the constraint of the derainer will collapse in the middle of the training process, which is caused by the difficulty and instability of the GAN manner. For fair comparison, all PSNR and SSIM scores reported in the main document are calculated on the RGB channels. The results of other methods are directly cited from the original papers or generated using the official models. For the results on datasets that the authors did not report or test, we retrain their models using the official code provided by the authors.

6. More Experiment Results

We present more experiment results on unsupervised deraining performance and generalization performance to further elucidate the effectiveness of the proposed CSUD.

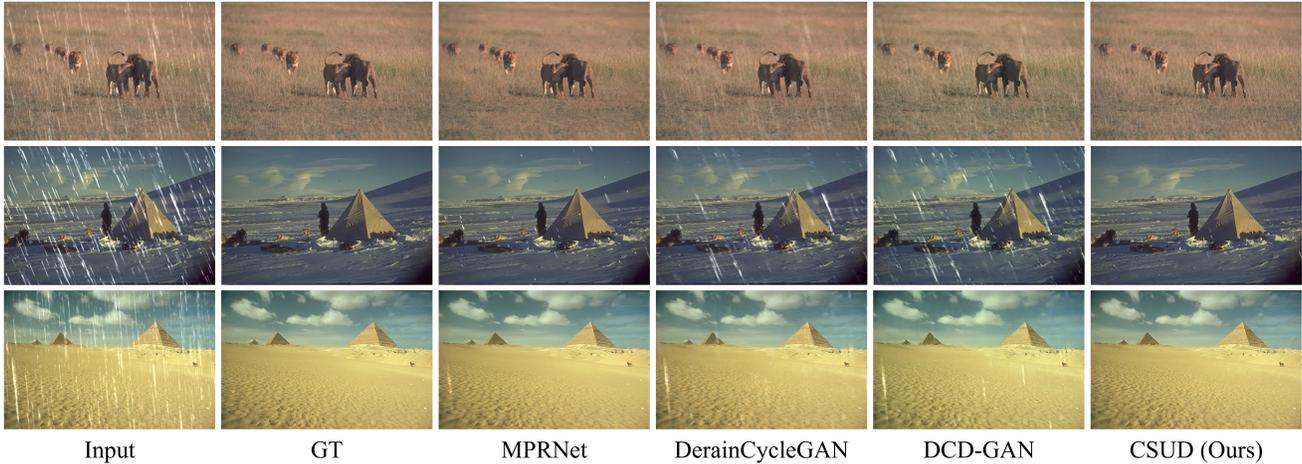


Figure 7. Qualitative deraining results on Rain100L [19] dataset.

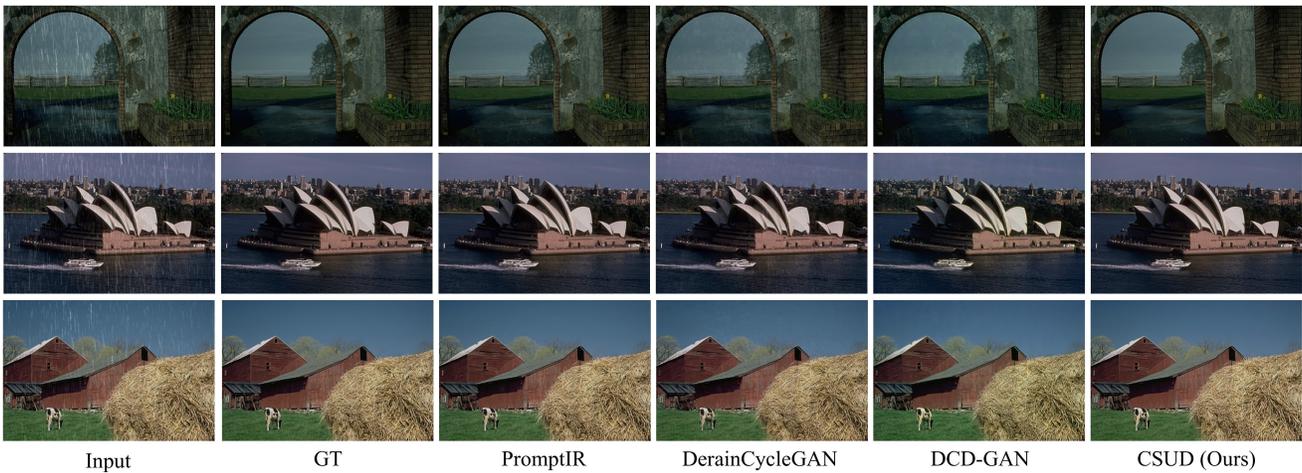


Figure 8. Qualitative deraining results on Rain12 [11] dataset.

6.1. Unsupervised Deraining Results

We provide additional visual comparisons on benchmark datasets in Fig. 7, Fig. 8, and Fig. 9. We compare our CSUD with several recent state-of-the-art unsupervised and supervised image deraining methods, including [2, 14, 18, 20, 21]. As shown in the figures, it can be seen that our CSUD achieves better results in removing rain streaks compared to other unsupervised methods and our CSUD preserves more texture details of image background. It is worth noting that there is a certain background color offset between the input and GT images of Rain800 dataset [9], however, our CSUD aims to preserve more color and texture details of image background while removing rain streaks, so our quantitative results in the main document which are are not the best.

6.2. Generalization Deraining Results

To validate the generalization capability of CSUD, we provide more additional visual comparisons with other un-

supervised and supervised deraining methods, including [2, 14, 18, 20, 22] in Fig. 12, Fig. 11, and Fig. 10. All methods are trained on synthetic datasets and tested on the unseen real-world datasets. Compared to other methods, our CSUD achieves better visual results in real-world scenarios, which demonstrates the excellent generalization capability of CSUD.

6.3. More Ablation Studies

Effect of CSUD framework on perceptual quality. To more comprehensively evaluate the performance of our CSUD in real world, we select 3 deraining baselines MPRNet [21], NeRD-Rain-S [4], and NAFNet [1], and we use additional perceptual quality metrics to test their supervised version and unsupervised version with our CSUD on 3 real-world datasets. The perceptual quality metrics includes full-reference metrics: LPIPS [26], DISTS [6] and no-reference metric: NIQE [25]. As shown in Tab. 2, our CSUD achieves

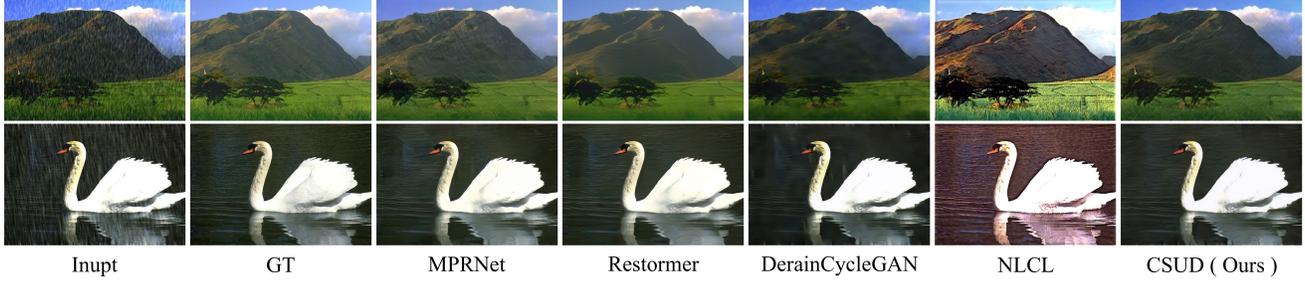


Figure 9. Qualitative deraining results on Rain800 [9] dataset.

Table 2. Quantitative perceptual quality comparisons of different deraining baselines with or without our methods.

Datasets	RealRain1K-L [10]	RealRain1K-H [10]	SPA-data [16]
Metrics	LPIPS ↓ / DISTs ↓ / NIQE ↓	LPIPS ↓ / DISTs ↓ / NIQE ↓	LPIPS ↓ / DISTs ↓ / NIQE ↓
MPRNet (Supervised)	0.355 / 0.279 / 8.872	0.424 / 0.314 / 8.231	0.159 / 0.125 / 7.946
MPRNet + CSUD (Unsupervised)	0.228 / 0.213 / 9.874	0.271 / 0.241 / 9.452	0.151 / 0.124 / 7.532
NeRD-Rain-S (Supervised)	0.341 / 0.298 / 7.132	0.445 / 0.339 / 6.679	0.167 / 0.131 / 7.151
NeRD-Rain-S + CSUD (Unsupervised)	0.336 / 0.298 / 7.104	0.441 / 0.338 / 6.489	0.160 / 0.130 / 7.079
NAFNet (Supervised)	0.308 / 0.285 / 7.150	0.416 / 0.328 / 6.722	0.152 / 0.124 / 7.108
NAFNet + CSUD (Unsupervised)	0.258 / 0.257 / 8.103	0.345 / 0.295 / 7.622	0.141 / 0.120 / 7.369

Table 3. Ablation experiments on the numbers of GANs. All models in the table are trained on Rain100L. SPA-Data and RealRain1K-L are used to evaluate the model’s generalization capability. Bold fonts indicate the highest metrics.

Num of GANs	Rain100L PSNR ↑ / SSIM ↑	RealRain1K-L PSNR ↑ / SSIM ↑	SPA-data PSNR ↑ / SSIM ↑
1	31.87 / 0.919	28.11 / 0.906	33.13 / 0.932
2	32.92 / 0.948	29.08 / 0.923	33.67 / 0.936
4	33.28 / 0.954	29.21 / 0.928	33.57 / 0.939

the best LPIPS and DISTs with all the 3 baselines, and NeRD-Rain-S with CSUD maintains best results for all the 3 perceptual metrics on all the 3 datasets. This shows that the derained image obtained by our method can obtain higher perceptual quality, and further demonstrates the effectiveness and the generalization ability of our methods.

Effect of the additional 3 GAN constraints. The introduction of the additional 3 adversarial constraints aims to enhance the training stability and improve the network’s performance. To validate the necessity, we respectively train the model with 1, 2, and 4 adversarial constraints, with results shown in Tab. 3. It is obvious that when 4 GAN constraints are used, the deraining performance and generalization ability of the network are the best, demonstrating the effectiveness of the additional 3 GAN constraints. Note that, during inference, only the derainer is used and our framework does not introduce any additional inference overhead.

Separation training of our framework. In order to further explore whether our unsupervised framework can train the generator and the derainer separately, we first train the generator separately and then train the derainer with

Table 4. Ablation experiments on separation training of CSUD framework. All models in the table are trained on Rain100L. Bold fonts indicate the highest metrics.

Training Strategy	Rain100L PSNR ↑ / SSIM ↑	RealRain1K-L PSNR ↑ / SSIM ↑	SPA-data PSNR ↑ / SSIM ↑
Separate Training	31.06 / 0.947	29.06 / 0.928	32.39 / 0.936
Joint Training	33.28 / 0.954	29.21 / 0.928	33.57 / 0.939

the pseudo-paired rain-clean image generated by the trained generator. As shown in Tab. 4, although it can still achieve good performance under separate training, its deraining performance and generalization ability have significantly decreased compared to joint training. Many components in our framework rely on the collaborative interaction to make derainer and generator mutually enhance each other. If the generator and derainer are trained separately, the SRloss for derainer and additional GAN constraints cannot be added to training process, and the generator cannot continuously generate pseudo-paired data, which will cause reduced constraints and performance degradation.

7. Discussion and Limitations

Like other image deraining methods, our method also face the same problem that it may mistakenly remove some background textures similar to the rain streaks in real rainy images, this shortcoming needs to be further improved. Additionally, our method can be widely applied in many applications such as autonomous vehicles and video surveillance. Therefore, one should be cautious of questionable results and avoid infringement of privacy or negative impact on society.



Figure 10. Qualitative generalization results on Internet-Data [17] dataset.

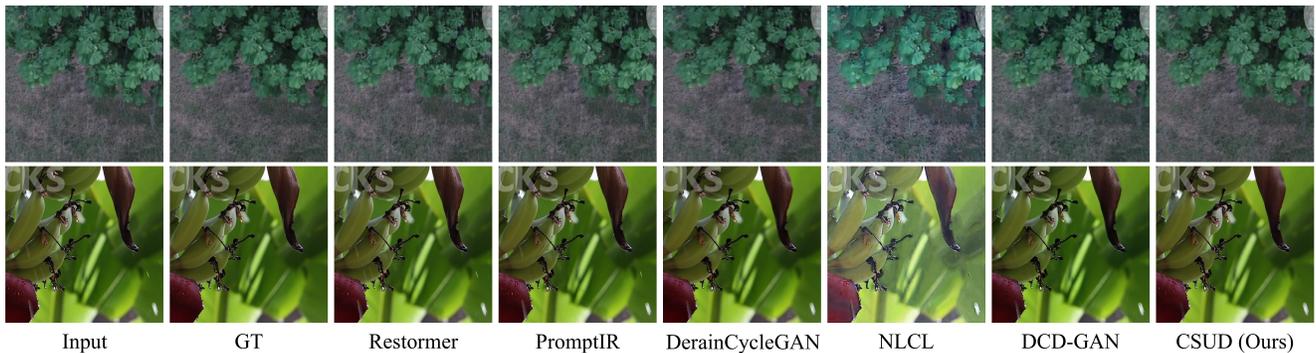


Figure 11. Qualitative generalization results on SPA-data [16] dataset.

References

- [1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European conference on computer vision*, pages 17–33. Springer, 2022. 3, 5
- [2] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2017–2026, 2022. 1, 5
- [3] Xiang Chen, Hao Li, Mingqiang Li, and Jinshan Pan. Learning a sparse transformer network for effective image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5896–5905, 2023.
- [4] Xiang Chen, Jinshan Pan, and Jiangxin Dong. Bidirectional multi-scale implicit neural representations for image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25627–25636, 2024. 1, 5
- [5] P Kingma Diederik. Adam: A method for stochastic optimization. (*No Title*), 2014. 4
- [6] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020. 5
- [7] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017. 1
- [8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 4
- [9] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3838–3847, 2019. 4, 5, 6
- [10] Wei Li, Qiming Zhang, Jing Zhang, Zhen Huang, Xinmei Tian, and Dacheng Tao. Toward real-world single image deraining: A new benchmark and beyond. *arXiv preprint arXiv:2206.05514*, 2022. 4, 6
- [11] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2736–2744, 2016. 4, 5
- [12] Di Lin, Xin Wang, Jia Shen, Renjie Zhang, Ruonan Liu, Miaohui Wang, Wuyuan Xie, Qing Guo, and Ping Li. Generative status estimation and information decoupling for image rain removal. *Advances in Neural Information Processing Systems*, 35:4612–4625, 2022. 1
- [13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An

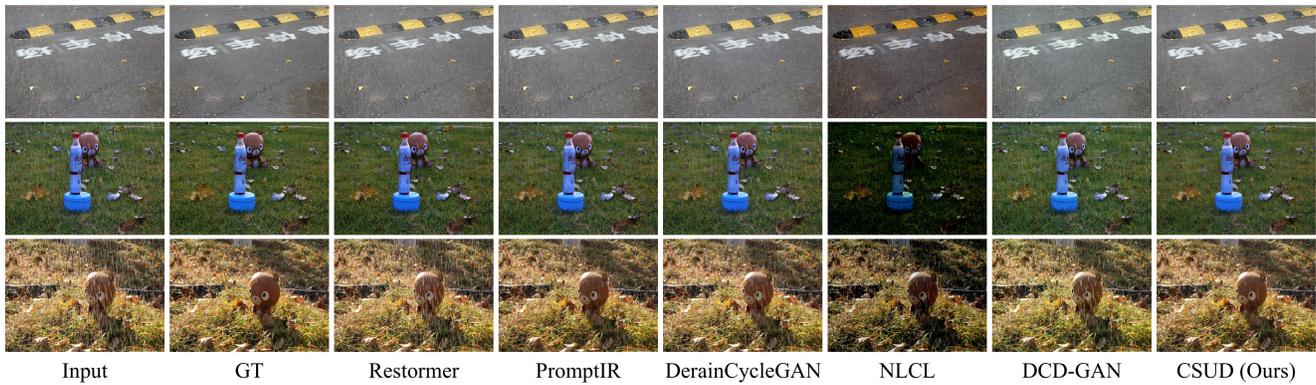


Figure 12. Qualitative generalization results on RainDS [15] dataset.

imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4

- [14] Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. *Advances in Neural Information Processing Systems*, 36, 2023. 5
- [15] Ruijie Quan, Xin Yu, Yuanzhi Liang, and Yi Yang. Removing raindrops and rain streaks in one go. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9147–9156, 2021. 4, 8
- [16] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12270–12279, 2019. 4, 6, 7
- [17] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3877–3886, 2019. 4, 7
- [18] Yanyan Wei, Zhao Zhang, Yang Wang, Mingliang Xu, Yi Yang, Shuicheng Yan, and Meng Wang. Deraincyclegan: Rain attentive cyclegan for single image deraining and rainmaking. *IEEE Transactions on Image Processing*, 30:4788–4801, 2021. 1, 4, 5
- [19] Wenhao Yang, Robby T Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1377–1393, 2019. 4, 5
- [20] Yuntong Ye, Changfeng Yu, Yi Chang, Lin Zhu, Xi-Le Zhao, Luxin Yan, and Yonghong Tian. Unsupervised deraining: Where contrastive learning meets self-similarity. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5821–5830, 2022. 5
- [21] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021. 5
- [22] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 1, 5
- [23] Fan Zhang, Shaodi You, Yu Li, and Ying Fu. Learning rain location prior for nighttime deraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13148–13157, 2023. 4
- [24] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 695–704, 2018. 1
- [25] Lin Zhang, Lei Zhang, and Alan C Bovik. A feature-enriched completely blind image quality evaluator. *IEEE Transactions on Image Processing*, 24(8):2579–2591, 2015. 5
- [26] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5
- [27] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3