Robust Adverse Weather Removal via Spectral-based Spatial Grouping

Supplementary Material

A. Overview

Our supplementary material offers further insights into the proposed method and provides in-depth discussions on topics that were not extensively covered in the main paper:

- Implementation details (Sec. B).
- Details of our architecture (Sec. C).
- Deep analysis of our proposed framework (Sec. D).
- Model capacity (Sec. E).
- Additional qualitative results (Sec. F).

B. Implementation details

Training procedure. We employ a progressive learning pipeline [15] with an initial batch size of 6 per GPU and a patch size of 128 , using 4 NVIDIA RTX 3090 GPUs. We use AdamW optimizer [6] and a cosine annealing schedule [7]. Following a progressive training pipeline [15], the patch size is adjusted to [128, 144, 216, 256, 288] at specific iteration milestones [150k, 280k, 370k, 424k, 472k], respectively. The model is trained for 900,000 iterations, beginning with an initial learning rate of $3e^{-4}$, which is progressively reduced to $1e^{-6}$ using a cosine annealing scheduler. These tensors serve as the initial states of the parameters and are optimized during training to achieve the best balance. For efficient training, we precompute the SVD feature for each image by performing SVD operations in advance. These precomputed SVD features are utilized throughout the training process without gradient computation.

Attention blocks. The number of blocks at each stage $L_{p\in\{1,2,3,4\}}$ is set to $\{4, 4, 6, 8\}$ blocks, divided into two types: SGTB-C and SGTB-S. For instance, stage 1 contains 4 blocks, with 2 SGTB-C and 2 SGTB-S blocks. The refinement block uses 4 transformer blocks from Restormer [15]. For in-group attention, α_1 is initialized as a tensor of ones with shape $\mathbb{R}^C$, and for cross-group attention, α_2 is initialized as a tensor of zeros with the same shape, where C represents the channel dimension. The number of groups g_p at each stage is set to $\{1, 2, 4, 8\}$ for SGTB-C and $\{256, 64, 16, 4\}$ for SGTB-S, where each value corresponds to stages 1 through 4. In SGTB-C, groups increase with channels, while in SGTB-S, they are scaled to maintain spatial size as resolution drops. Channel size C is set to 36.

Objective function. The Pearson correlation [1] loss brings the patch-level linear correlations by capturing the relative luminance dynamics inherent in the ground-truth [11]. The detailed correlation loss, $\mathcal{L}_{cor}$, is following as,

$$\rho\left(I^{hq}, I^{gt}\right) = \frac{\sum_{i=1}^{3HW}\left(I_i^{hq} - \overline{I}^{hq}\right)\left(I_i^{gt} - \overline{I}^{hq}\right)}{3HW \cdot \sigma(I^{hq}) \cdot \sigma(I^{gt})}, \quad (1)$$

$$\mathcal{L}_{\text{cor}} = \frac{1}{2}\left(1 - \rho\left(I^{gt}, I^{gt}\right)\right), \quad (2)$$

where $I_i^{\{t\}}$ represents the i-th pixel of image, $\overline{I}^{\{\bullet\}}$ and $\sigma(I^{\{\bullet\}})$ indicate, respectively, the sequence's mean pixel value and its standard deviation. The total objective function we used is as:

$$\mathcal{L} = \mathcal{L}_{rec} + \beta\mathcal{L}_{cor}, \quad (3)$$

where β is 1.

C. Details of our architecture

In this section, we provide a detailed explanation of our proposed architecture. First, we describe the Sobel operator and SVD filter, which perform spectral decomposition on the input image. Next, we introduce the Sobel refinement block and the SVD refinement block, which further refine the extracted spectral features. Finally, we explain the feature-grouped attention layer in detail, a key component of the SGTB for effective attention processing.

Sobel operator and SVD filter. The Sobel operator [2] computes vertical and horizontal gradients using two kernels (K_x, K_y). By applying this operator, fine gradient variations are highlighted, revealing high-frequency components that capture degradation patterns and texture details. This process helps identify degradation characteristics while providing valuable contextual information about the overall image structure.

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

Before applying the Sobel operator, we use a 3×3 convolution layer instead of the traditionally used Gaussian filter to reduce noise and align the channel dimensions.

SVD filter [9, 10, 14] analyzes the distribution of low-frequency information across pixels to identify degradation patterns. It decomposes an image into singular value components, where larger singular values correspond to dominant structures, while smaller values capture finer details. By detecting large-scale degradation, such as snow or raindrops, SVD effectively highlights regions that obscure broad areas without significant texture or fine details.

The detailed SVD filter operation is follows. At first, we apply padding to the given degraded image I_D with size w and create a padded image I_{pad}, which allows the calculation near the edges. For each pixel location (i, j) in I_D, a corresponding block B_{ij} of size $(2w + 1) \times (2w + 1)$ is extracted from I_{pad}. The block B_{ij} is defined as:

$$B_{ij} = I_{pad}[i : i + (2w + 1), j : j + (2w + 1)]. \quad (5)$$

SVD is performed, resulting in a decomposition $B_{ij} = U\Sigma V^T$, where $\Sigma = \text{diag}(s_1, s_2, \ldots, s_n)$ contains the singular values s_k for the block. To calculate the degree for each block, the sum of the top l singular values is divided by the total sum of singular values in the block. This ratio, defined as the degree, $D_{i,j}$, is given by

$$D_{i,j} = \frac{\sum_{k=1}^{l} s_k}{\sum_{k=1}^{n} s_k}. \quad (6)$$

After calculating the degree for all blocks, normalization is applied to scale $D_{i,j}$ values between 0 and 1. The normalized map value, *i.e.,* the SVD feature, F_{SVD} is obtained by

$$F_{\text{SVD}}(i, j) = \frac{D_{i,j} - \min(D)}{\max(D) - \min(D)}. \quad (7)$$

Sobel and SVD refinement block. The Sobel refinement block consists of a convolution layer, a linear layer, and a learnable parameter with the same size as the feature channels. The process referred to as feature reorganization in the main paper involves passing the feature through a 3×3 convolution layer, reshaping it, applying a linear layer, and then performing a softmax operation along the channel dimension before multiplying it with the learnable parameter. This process refines the Sobel feature based on local region information.

The SVD refinement block consists of convolution layers, feature reorganization, a deformable convolution layer, and a learnable parameter, which refine the low-frequency components with I_D. First, F_{SVD} is interpolated to the size of the main feature, F_p. The interpolated feature is then concatenated with the feature extracted from the input image and passed through a 3×3 convolution layer followed by a deformable convolution layer. Afterward, similar to the Sobel refinement block, the feature passes through a 3×3 convolution layer, is reshaped, undergoes a linear transformation, and is then multiplied by the learnable parameter. This process enhances the SVD feature with the context of the input image.

Feature-grouped attention in SGTB. We describe the internal structure of the feature-grouped attention layer, one of the key components of SGTB. First, using the grouping-mask M_p obtained from the mask generator (where p denotes the stage index), we perform grouping the intermediate feature. The grouped features are then processed through a projection layer to generate query, key, and value representations. The projection layer consists of a 1×1 convolution layer followed by a 3×3 depth-wise convolution. Once the query, key, and value are obtained, as shown in Eq. (5) of the main paper, both in-group attention and cross-group attention are performed simultaneously. In the case of SGTB-C, the attention results for the m-th group, A^m_{in} and A^m_{cross}, have dimensions of $C \times C$. In contrast, for SGTB-S, A^m_{in} and A^m_{cross} have dimensions of $\frac{HW}{g_p} \times \frac{HW}{g_p}$. Finally, the separate attention results are merged and passed through a 1×1 convolution layer to produce the final output. Here, H and W are the spatial dimensions of the feature and g_p is the total number of groups in stage p.

Table A. A toy experiment for window size of SVD filter.

Window Size	PSNR	SSIM
1	30.87	0.9235
3	<u>31.14</u>	0.9241
5	**31.29**	<u>0.9258</u>
7	31.13	**0.9259**

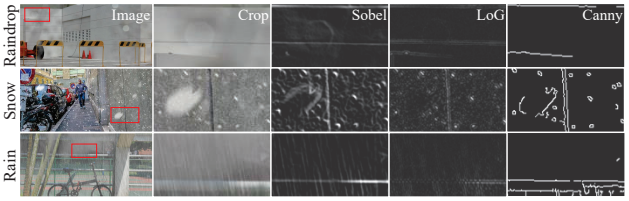

Figure A. Visualization of outputs of various edge detectors.

D. Deep analysis of our proposed method

SVD window. In our approach, we applied an SVD filter to patches of a specific size, using only the pixel values within each patch. To determine the optimal patch size for this filter, we conducted a simple toy experiment. In this experiment, we kept all conditions consistent, including the torch seed, except for varying the window size. Each case was trained on a single GPU, and for validation, we evaluated the model based on 100k iterations, using the raindrop metric. The results in Tab. A showed that a window size of 5 yielded the highest PSNR, and the SSIM was second-best with a difference of only 0.0001. Based on these findings, we selected the window size, w, of 5 for the final model.

Options for edge detector. We experimented with various edge detectors, including Sobel, Canny, and Laplacian of Gaussian (LoG). Rather than relying solely on quantitative metrics, we visually inspected the filtered outputs for practical effectiveness, as shown in Fig. A. While second-order methods like LoG highlighted background edges, they often missed fine weather-induced details. In contrast, first-order methods such as Sobel were more robust under adverse conditions, leading us to adopt Sobel.

Effect of group number. We conduct a toy experiment to compare our proposed number of groups with two constant group configurations: $\{4, 4, 4, 4\}$ for SGTB-C and $\{64, 64, 64, 64\}$ for SGTB-S. To ensure fair comparison, we adopt the same GPU and training configurations used in the main experiments. We limit the training to 100k iterations, as this setup is intended as a lightweight analysis. The results, shown in Tab. B, demonstrate that our method outperforms both constant configurations in terms of accuracy. Furthermore, our method achieves this while consuming the least training memory among all compared settings.

Table B. Ablations with a single GPU, 100k trained.

Method	Rainfog		Snow		Raindrop		Avg	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Ours {1,2,4,8} / {256, 64, 16, 4}	**26.95**	**0.884**	**28.63**	**0.878**	**31.24**	0.925	**28.94**	**0.896**
{4,4,4,4} / {64, 64, 64, 64}	26.30	0.880	28.30	0.877	31.08	**0.926**	28.56	0.894
{1,2,4,8} / {64, 64, 64, 64}	26.55	0.880	28.10	0.877	30.93	0.923	28.53	0.893
{4,4,4,4} / {256, 64, 16, 4}	26.90	**0.884**	28.16	0.876	31.18	0.925	28.75	0.895

Grouping-mask visualization. For group-wise attention, we generated a grouping-mask for spatial grouping. We provide visual results to illustrate the information conveyed by the mask. As shown in Fig. B, the mask groups areas with similar characteristics. In the first row, the mask effectively groups areas obscured or blurred by raindrops. The stairs, blurred by water droplets, are not clearly visible in the original image but are distinctly highlighted in the mask. In the second row, the rain streak is clearly noticeable, and the areas obscured by fog are well distinguished from the sky. In the third row, the snowflake particles are effectively separated from the background. We performed group-wise attention, allowing the attention mechanism to focus on the relevant regions for restoration.

No-reference perceptual metrics. To further demonstrate the superior quality of the clean images produced by our method, we additionally evaluate them using non-reference image quality assessment metrics. We employed MUSIQ [3] and CLIP-IQA+ [13], two widely used non-reference metrics in recent studies. As shown in Tab. C, our method achieves higher MUSIQ and CLIP-IQA+ scores, further confirming perceptual gains.

E. Model capacity

To provide a comprehensive evaluation, we compare our method with state-of-the-art models for adverse weather removal, considering the efficiency and performance metrics. The results are analyzed in terms of PSNR and SSIM to highlight the effectiveness of our approach in Tab. D. As shown in the table, our model achieves the best results while maintaining a highly competitive model size and inference time.

F. Additional qualitative results

We show additional qualitative comparisons to showcase the effectiveness of our method on weather-degraded datasets. Specifically, the samples for raindrop removal can be found in Fig. C and D, rain removal in Fig. E and F, and snow removal in Fig. G and H. These examples are sourced from the RainDrop [8], Outdoor-Rain [4], and Snow100K-L [5] datasets, respectively. Additionally, we trained our model on the real-world WeatherStream dataset [16] and evaluated its performance on rain, fog, and snow removal within the same dataset, as shown Fig. J. Our method demonstrates superior performance in restoring degraded

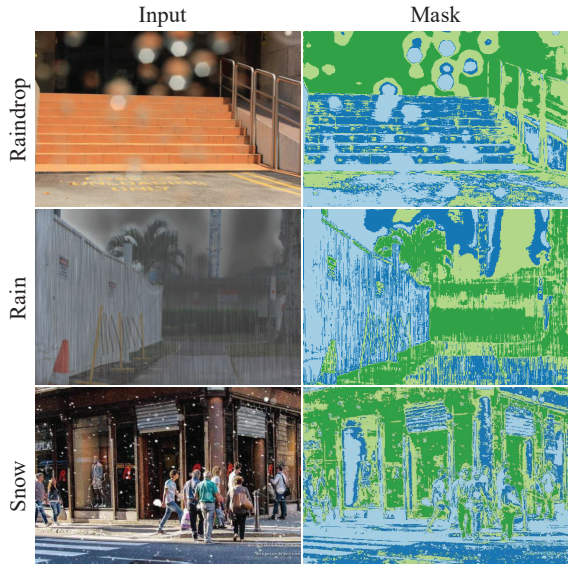

Figure B. Visualization of grouping-mask. The left column is the input image, and the right column is the mask obtained from the mask generator. From top to bottom, the results correspond to raindrop, rain, and snow. The mask, divided into four groups, can effectively group areas with similar features. The same group has the same color.

Table C. Quantitative comparisons on the All-weather dataset with non-reference metrics.

Method	Rainfog		Snow		Raindrop		Avg	
	MUSIQ	IQA+	MUSIQ	IQA+	MUSIQ	IQA+	MUSIQ	IQA+
Histo. [11]	70.16	0.6145	65.63	0.5894	70.66	0.6530	68.82	0.6190
Ours	**70.46**	**0.6452**	**65.91**	**0.6171**	**71.85**	**0.6580**	**69.41**	**0.6401**

Table D. Model size and performance. The inference time refers to the processing time for a 256×256 image.

	TransWeather [12]	WGWS [17]
PSNR	29.44	30.62
SSIM	0.901	0.920
Param. (M)	38.05	5.19
Inference time (s)	0.14	0.14
	Histoformer [11]	Ours
PSNR	32.43	32.63
SSIM	0.936	0.939
Param. (M)	16.61	16.65
Inference time (s)	0.72	0.86

images closer to the ground truth compared to other methods. In particular, as illustrated in Fig. F, the restored image effectively reproduces the color tone of the cropped region from the original ground truth. Moreover, we provide additional real world degradation images and their recovered images. As shown in Fig. I, our method effectively removes snow particles and restores the background in real snow environments, as our model does in synthetic scenarios.

References

[1] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009. 1

[2] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2): 358–367, 1988. 1

[3] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021. 3

[4] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1633–1642, 2019. 3, 5

[5] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE Transactions on Image Processing*, 27(6): 3064–3073, 2018. 3, 6

[6] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[7] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[8] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018. 3, 5

[9] Filip Sroubek, Jan Kamenicky, and Yue M Lu. Decomposition of space-variant blur in image deconvolution. *IEEE signal processing letters*, 23(3):346–350, 2016. 1

[10] Bolan Su, Shijian Lu, and Chew Lim Tan. Blurred image region detection and classification. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1397–1400, 2011. 1

[11] Shangquan Sun, Wenqi Ren, Xinwei Gao, Rui Wang, and Xiaochun Cao. Restoring images in adverse weather conditions via histogram transformer. In *European Conference on Computer Vision*, pages 111–129. Springer, 2025. 1, 3

[12] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2353–2363, 2022. 3

[13] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2555–2563, 2023. 3

[14] Huimei Xiao, Wei Lu, Ruipeng Li, Nan Zhong, Yuileong Yeung, Junjia Chen, Fei Xue, and Wei Sun. Defocus blur detection based on multiscale svd fusion in gradient domain. *Journal of Visual Communication and Image Representation*, 59:52–61, 2019. 1

[15] 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

[16] Howard Zhang, Yunhao Ba, Ethan Yang, Varan Mehra, Blake Gella, Akira Suzuki, Arnold Pfahnl, Chethan Chinder Chandrappa, Alex Wong, and Achuta Kadambi. Weatherstream: Light transport automation of single image deweathering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13499–13509, 2023. 3, 7

[17] Yurui Zhu, Tianyu Wang, Xueyang Fu, Xuanyu Yang, Xin Guo, Jifeng Dai, Yu Qiao, and Xiaowei Hu. Learning weather-general and weather-specific features for image restoration under multiple adverse weather conditions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21747–21758, 2023. 3

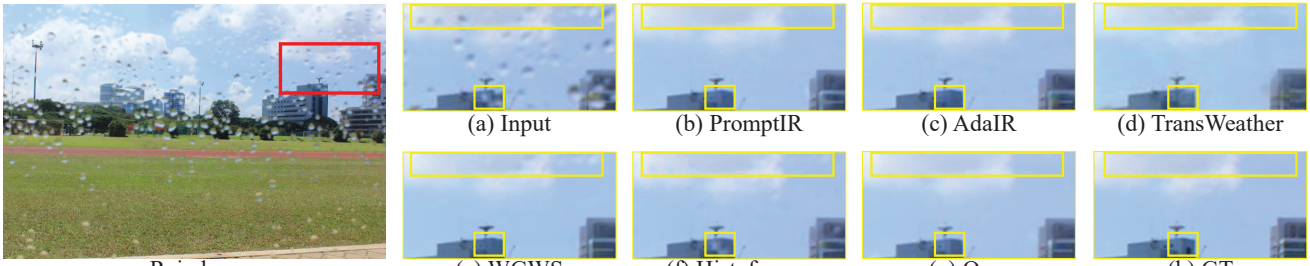

Figure C. Qualitative results of raindrop removal on RainDrop [8] dataset. Zoom for better view.

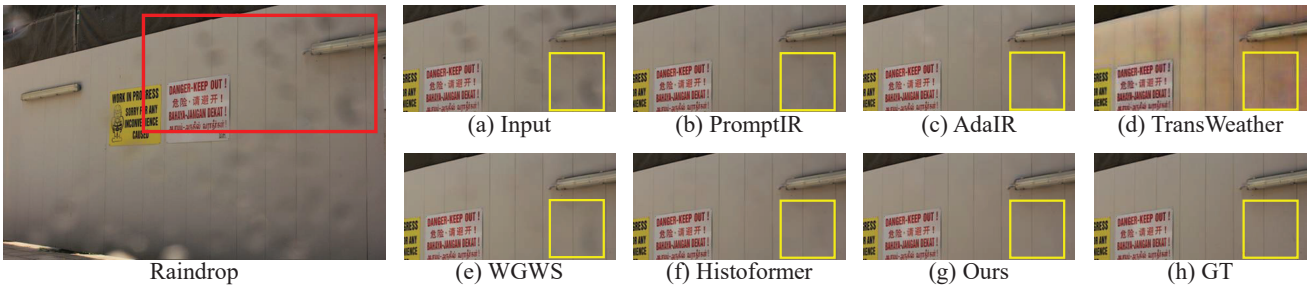

Figure D. Qualitative results of raindrop removal on RainDrop [8] dataset. Zoom for better view.

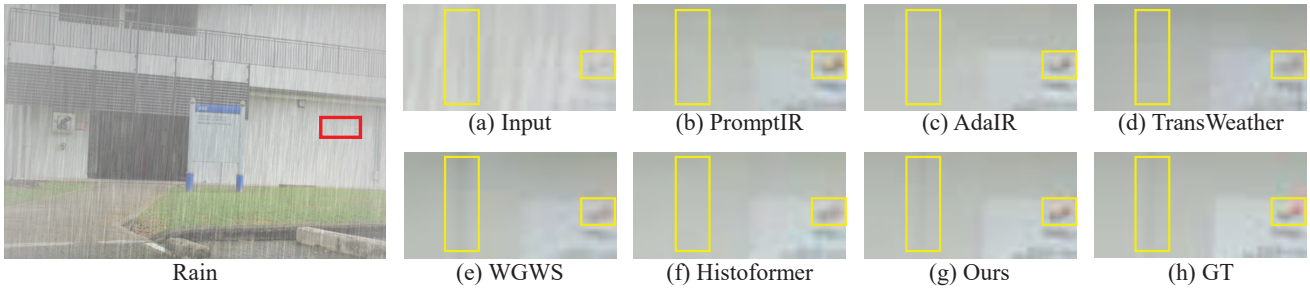

Figure E. Qualitative results of deraining on Outdoor-rain [4] dataset. Zoom for better view.

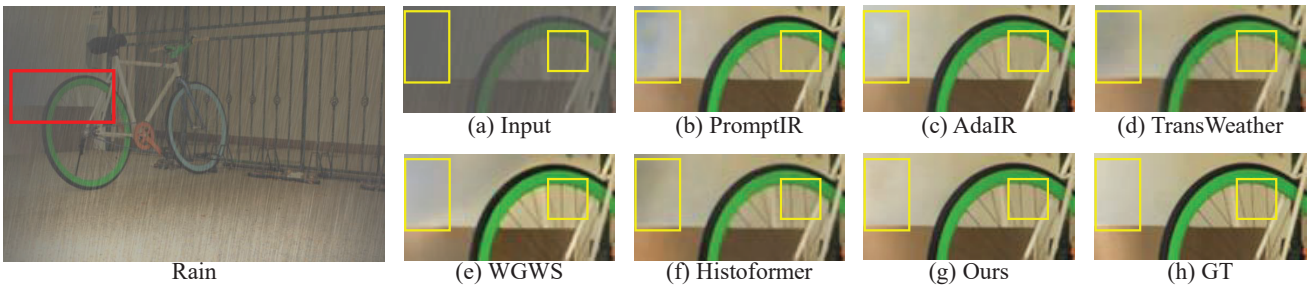

Figure F. Qualitative results of deraining on Outdoor-rain [4] dataset. Zoom for better view.

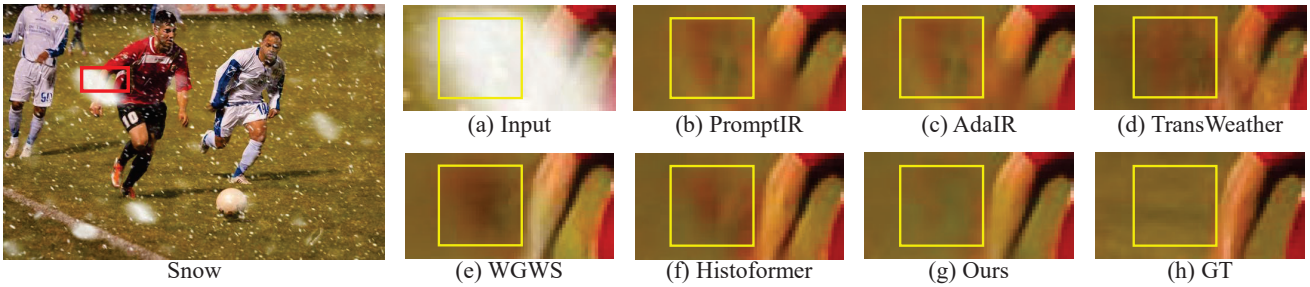

Figure G. Qualitative results of desnowing on Snow100K-L [5] dataset. Zoom for better view.

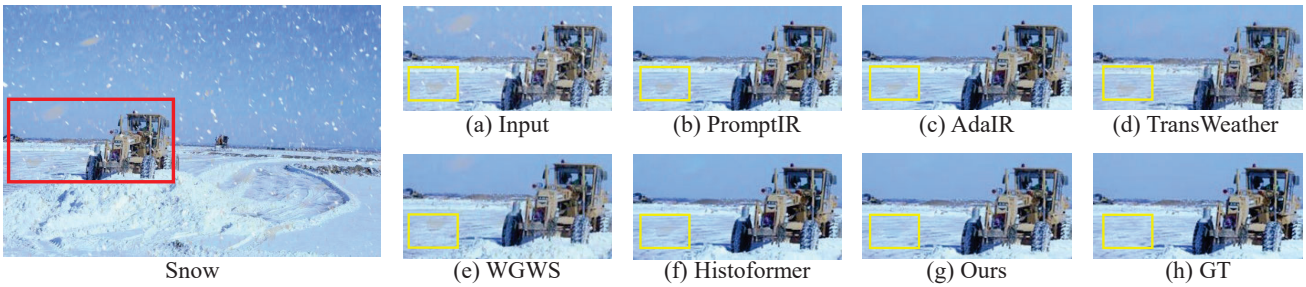

Figure H. Qualitative results of desnowing on Snow100K-L [5] dataset. Zoom for better view.

Figure I. Visual comparison of real-world snowfall, based on a model trained on the synthetic dataset and applied to real-world scenarios. Zoom for better view.

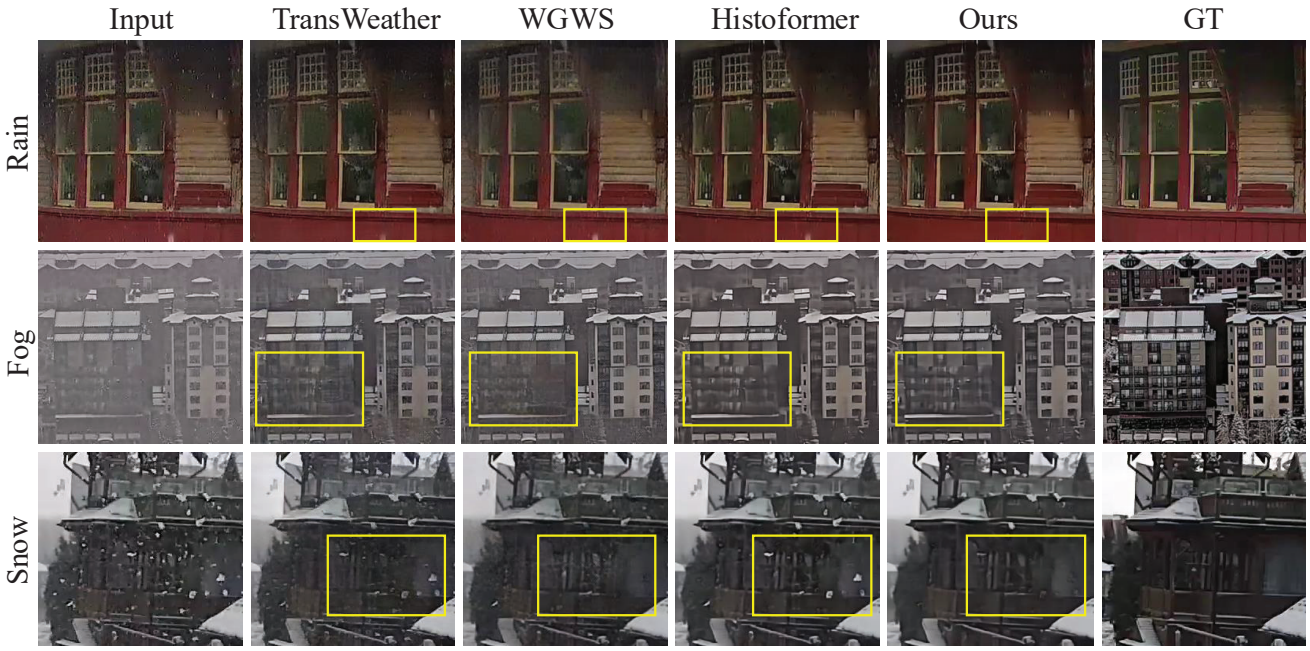

Figure J. Qualitative results for rain, fog, and snow removal on the real-world WeatherStream dataset [16]. Zoom for better view.