Supplementary Material of Towards Better De-raining Generalization via Rainy Characteristics Memorization and Replay

Anonymous Author(s)

I. Framework Complexity Analysis

We analyze how the cost in terms of parameters, computation, and time scales with the number of integrated datasets N. Let D_n denote the n-th dataset, with M_n image pairs. The CLGID training framework consists of three core stages at each step: training a GAN G_n on D_n, generating replay data $\hat{D}_n$ from earlier GANs $\{G_1, \cdots, G_{n-1}\}$, and training the de-raining network on $D_n \cup \hat{D}_n$.

GAN Complexity Analysis

Parameter cost: Each new dataset adds one GAN. If a single GAN contains P_G parameters, total parameter cost scales linearly:

$$P_{\text{GAN}} = N \cdot P_G$$

Training FLOPs and time: Assume GANs are trained for E_G epochs with per-batch (B_G) FLOPs cost F_G^{Train} and time cost t_G^{Train}. Then for each stage:

$$\text{FLOPs}_{\text{GAN}}^{(n)} = E_G \cdot \frac{M_n}{B_G} \cdot F_G^{\text{Train}},$$

$$T_{\text{GAN}}^{(n)} = E_G \cdot \frac{M_n}{B_G} \cdot t_G^{\text{Train}}.$$

Accumulated over N datasets:

$$\text{FLOPs}_{\text{GAN}} = \sum_{n=1}^{N} E_G \cdot \frac{M_n}{B_G} \cdot F_G^{\text{Train}},$$

$$T_{\text{GAN}} = \sum_{n=1}^{N} E_G \cdot \frac{M_n}{B_G} \cdot t_G^{\text{Train}}.$$

Replay FLOPs and time: At stage n, replay dataset $\hat{D}_n$ is generated to match the size of the current dataset T_n, i.e., $|\hat{D}_n| = M_n$. The samples are drawn by uniformly sampling from each of the $n-1$ GANs. Each sample requires FLOPs F_R and time T_R:

$$\text{FLOPs}_{\text{Replay}}^{(n)} = M_n \cdot F_R, \quad T_{\text{Replay}}^{(n)} = M_n \cdot T_R.$$

Accumulated over N datasets:

$$\text{FLOPs}_{\text{Replay}} = \sum_{n=1}^{N} M_n \cdot F_R, \quad T_{\text{Replay}} \sum_{n=1}^{N} M_n \cdot T_R$$

De-raining Network Complexity Analysis

Parameter cost: The de-raining backbone remains fixed throughout training. Let it have P_D parameters:

$$P_{\text{D-net}} = P_D \text{ (constant w.r.t. } N). \tag{1}$$

Training FLOPs and time: Let F_D^{Train} and t_D^{Train} be the FLOPs and time cost for per-batch (B_D) forward–backward pass. With E_D training epochs:

$$\text{FLOPs}_{\text{D-net}}^{(n)} = E_D \cdot \frac{M_n}{B_D} \cdot F_D^{\text{Train}},$$

$$T_{\text{D-net}}^{(n)} = E_D \cdot \frac{M_n}{B_D} \cdot t_D^{\text{Train}}.$$

Accumulated over N datasets:

$$\text{FLOPs}_{\text{D-net}} = \sum_{n=1}^{N} E_D \cdot \frac{M_n}{B_D} \cdot F_D^{\text{Train}},$$

$$\text{T}_{\text{D-net}} = \sum_{n=1}^{N} E_D \cdot \frac{M_n}{B_D} \cdot t_D^{\text{Train}}.$$

II. The proof of the total replay cost in GAN-replayed data reuse

Suppose the dataset sizes are upper bounded: $M_n \leq M_{\max}$, and denote $M_{\min} = \min_n M_n > 0$. Then we can derive an upper bound:

$$\Delta_{i,n} \leq \max\left(0, \frac{M_{\max}}{n-1} - \frac{M_{\min}}{n-2}\right).$$

More importantly, the dominant term across all n is:

$$\Delta_{n-1,n} = \frac{M_n}{n-1} \leq \frac{M_{\max}}{n-1}.$$

Therefore, the total cost satisfies:

$$C_N \leq \sum_{n=2}^{N} \left((n-2) \cdot \varepsilon_n + \frac{M_{\max}}{n-1}\right),$$

where

$$\varepsilon_n = \max\left(0, \frac{M_{\max}}{n-1} - \frac{M_{\min}}{n-2}\right).$$

Note that ε_n vanishes for sufficiently large n. Specifically, solving

$$\frac{M_{\max}}{n-1} \leq \frac{M_{\min}}{n-2} \Longleftrightarrow \frac{M_{\max}}{M_{\min}} \leq \frac{n-1}{n-2},$$

gives a threshold

$$N_0 = \left\lceil 1 + \frac{M_{\max}}{M_{\min} - M_{\max}} \right\rceil.$$

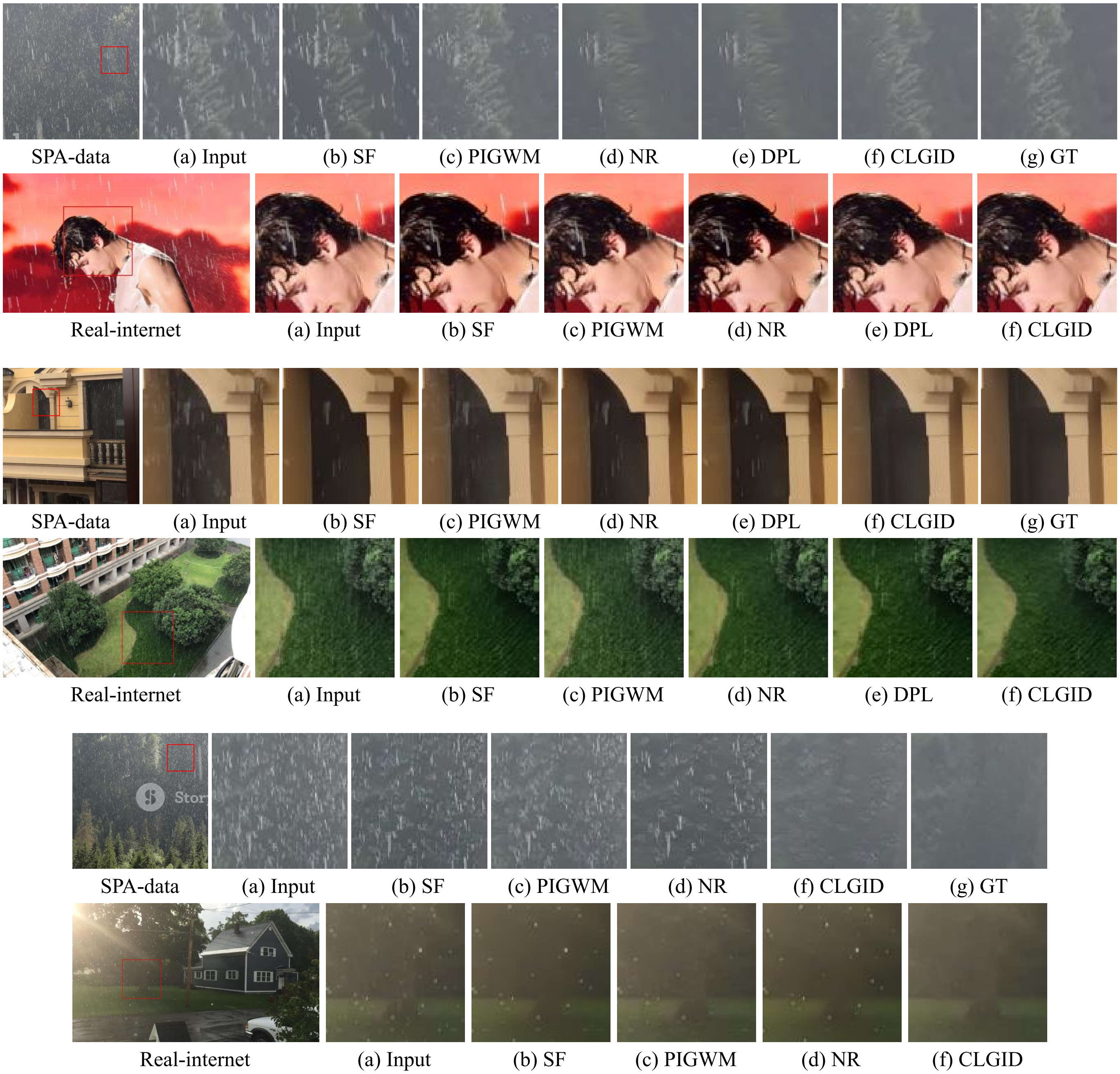


Fig. 1: Visual quality comparisons of different methods on SPA-data [1] and Real-internet [1]. From top to bottom: MFDNet [2], Restormer [3], and MPRNet [4] are used as the de-raining networks. Ground truth is available for SPA-data but not for Real-internet. Note that DPL is not applicable to non-transformer-based networks such as MPRNet.

above which $\varepsilon_n = 0$ for all $n \geq N_0$. Thus,

$$\sum_{n=2}^{N}(n-2)\varepsilon_n \leq \sum_{n=2}^{N_0}(n-2)\varepsilon_n \triangleq C_0,$$

where C_0 is a finite constant independent of N. Therefore, we can treat the first summation as a constant. The second term forms a harmonic sum:

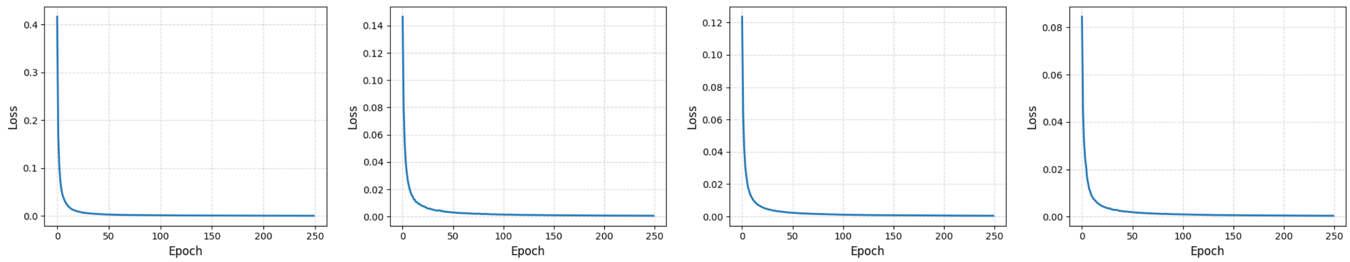
$$\sum_{n=2}^{N}\frac{M_{\max}}{n-1} = M_{\max}\sum_{k=1}^{N-1}\frac{1}{k} = M_{\max}\cdot H_{N-1},$$

where $H_{N-1} \leq \ln(N-1)+1$. Therefore,

$$C_N = \mathcal{O}(M_{\max}\log N).$$

III. Quantitative Comparisons of Different Methods

In addition to quantitative results, we also provide qualitative assessments of various methods on SPA-data [1] and Real-internet [1], after training on the 1400-1200M-100H-100L sequence, as shown in Fig. 1. We can observe that other



(a) Dataset 1 (b) Dataset 2 (c) Dataset 3 (d) Dataset 4

Fig. 2: Training loss curve under the sequence 1400-1200M-100H-100 using MPRNet [4].

methods struggle to eliminate heavy rain streaks and those resembling the background's texture. Some artifacts persist in the de-raining outcomes, and the background details appear blurred. In contrast, our CLGID yields the most visually appealing results.

IV. Visualization of Loss Convergence

To verify the convergence behavior of the proposed framework, we provide the training loss curve under a representative setting (MPRNet + 1400-1200M-100H-100L), as shown in Fig. 2. The curve clearly demonstrates that the training process exhibits smooth and stable convergence. Within each dataset training phase, the loss decreases steadily, showing consistent optimization. Moreover, at the transition between datasets, there is no sign of abrupt increase, conflict, or collapse in the loss, indicating that the framework handles dataset shifts in a stable manner.

References

[1] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[2] Q. Wang, K. Jiang, Z. Wang, W. Ren, J. Zhang, and C.-W. Lin, "Multi-scale fusion and decomposition network for single image deraining," *IEEE Transactions on Image Processing*, 2023.

[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[4] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.