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

Anonymous Author(s)

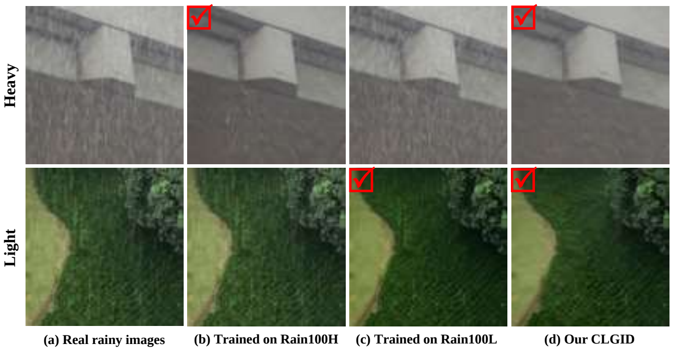
Abstract—**Current image de-raining methods primarily learn from a limited dataset, leading to inadequate performance in varied real-world rainy conditions. To tackle this, we introduce a new framework that enables networks to progressively expand their de-raining knowledge base by tapping into a growing pool of datasets, significantly boosting their adaptability. Drawing inspiration from the human brain's ability to continually absorb and generalize from ongoing experiences, our approach borrow the mechanism of the complementary learning system. Specifically, we first deploy Generative Adversarial Networks (GANs) to capture and retain the unique features of new data, mirroring the hippocampus's role in learning and memory. Then, the de-raining network is trained with both existing and GAN-synthesized data, mimicking the process of hippocampal replay and interleaved learning. Furthermore, we employ knowledge distillation with the replayed data to replicate the synergy between the neocortex's activity patterns triggered by hippocampal replays and the pre-existing neocortical knowledge. This comprehensive framework empowers the de-raining network to accumulate knowledge from various datasets, continually enhancing its performance on previously unseen rainy scenes. Our testing on three benchmark de-raining networks confirms the framework's effectiveness. It not only facilitates continual knowledge accumulation across six datasets but also surpasses state-of-the-art methods in generalizing to new real-world scenarios.**

Index Terms—**Image de-raining, deep learning, generalization, knowledge accumulation.**

I. Introduction

Single image de-raining, which seeks to eliminate rain streaks from images to reveal their clean versions, is pivotal for enhancing the efficacy of subsequent vision tasks like classification and detection [1–7]. Despite the advancements achieved by deep learning-based de-raining methods [8–12], a critical shortcoming remains: their reliance on learning specific de-raining patterns from a limited set of rainy images. This approach results in underwhelming performance when applied in varied real-world conditions, due to the inability to fully represent the complexity of real-world rain distribution, as illustrated in Fig 1. To surmount this challenge, it is essential to develop de-raining methods capable of continually expanding their knowledge by learning from an ever-increasing collection of de-raining datasets. This strategy enables the networks to significantly improve their adaptability and performance in real-world scenarios, addressing the partial coverage issue of current specific de-raining mappings.

One potential solution is to integrate newly acquired data with existing data and retrain the network from scratch. However, this approach requires retraining for each new dataset, and as the combined dataset grows, the retraining costs escalate, making this method impractical due to the significant computational expense. Alternatively, the de-raining network could be sequentially trained on newly acquired data. However, this approach is prone to catastrophic forgetting [15], which occurs due to interference between new and previously learned knowledge, resulting in an ineffective accumulation of de-raining knowledge.

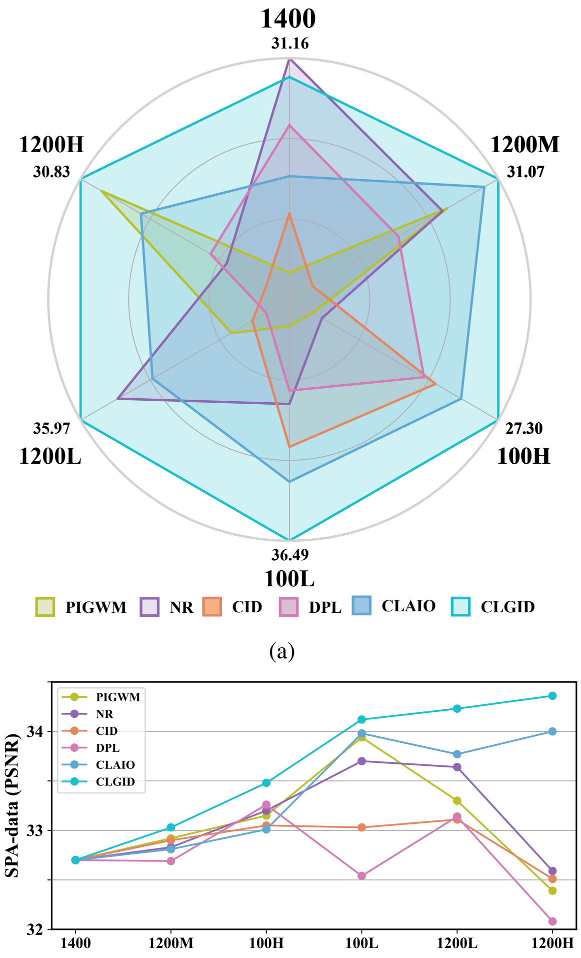


(a) Real rainy images (b) Trained on Rain100H (c) Trained on Rain100L (d) Our CLGID

Fig. 1: Visual comparison between a de-raining network (exemplified by the Multi-scale Fusion De-raining Network (MFDNet)[13]) trained on a fixed de-raining dataset and our proposed continual de-raining framework (CLGID), which accumulates de-raining knowledge from a sequence of de-raining datasets. While MFDNet trained on a fixed dataset can handle only specific types of real-world rain streaks (i.e., Rain100H[14] → Heavy, Rain100L[14] → Light), our CLGID is capable of addressing diverse real-world rainy scenes.

To address this issue, recent efforts have been focused on efficiently gathering de-raining insights from data streams. These initiatives include various strategies, such as adjusting model parameter weights [16], segregating model parameters [17], and learning prompts specific to the task [18]. However, these methods continue to face challenges in memory capacity and generalization. As illustrated in Fig 2, the introduction of additional datasets leads to a saturation of memory capacity, which in turn limits the enhancement of the network's ability to generalize.

On the other hand, humans have an exceptional capacity to continuously learn and remember various events, extracting statistical patterns from these events to develop the ability to generalize to new situations. The complementary learning system of the human brain, involving the hippocampus and neocortex, is pivotal in this cognitive process [15, 24, 25]. Inspired by this remarkable capability, an intriguing question arises: can we draw inspiration from the human brain's complementary learning system, which facilitates ongoing memory of events and generalization across these memories,



(a)

(b)

Fig. 2: (a) Comparison of memory performance across each of the six datasets after training on a stream of six datasets [14, 19, 20]: 1400-1200M-100H-100L-1200L-1200H. (b) Generalization performance variance on ***unseen real-world SPA-data*** [21] during training on a stream of six datasets. We compare the SOTA methods PIGWM [16], NR [17], CID [22], DPL [18], and CLAID [23] with our proposed CLGID, all using MFDNet [13] as the de-raining backbone.

to overcome existing challenges of the image de-raining?

In this paper, we draw inspiration from the human brain's Complementary Learning system to introduce a new Generalized Image De-raining framework (CLGID). Specifically, the complementary learning system comprises of the hippocampus and the neocortex. The hippocampus is responsible for learning and storing unique perceptions of events [26]. It then replays these memories, mixed with new events, to the neocortex. Through a cycle of replaying hippocampal memories and interspersed learning, the neocortex harmonizes these memories with existing neocortical knowledge [27], gradually extracting structured insights and developing the ability to generalize to new scenarios.

Mirroring this, we use GANs to mimic the hippocampus by learning and store the rainy characteristics of each dataset. The de-raining network, acting as the neocortex, is then trained on a mix of GAN-generated memories and current data. This method replicates the hippocampal to neocortical replay and interleaved learning process, fostering the network's ability to generalize across data. Additionally, we incorporate knowledge distillation with replayed data to ensure the neocortical activity patterns, triggered by hippocampal replays, align with pre-existing neocortical knowledge. Extensive experiments on three representative de-raining networks [13, 28, 29] confirm that CLGID effectively preserves memory across six datasets and constantly improves generalization to unseen real-world images, outperforming existing methods. Fig. 2 presents results based on MFDNet [13].

Additionally, recent discoveries in cognitive science [30, 31] bolster the theory of the complementary learning system, suggesting the neocortex can swiftly assimilate structured knowledge when new events closely resemble past ones. Motivated by these insights, we propose a pattern similarity-based training acceleration algorithm to enhance our CLGID framework. Essentially, we evaluate the similarity in rain patterns between the new dataset and GAN-generated memories before beginning training. Depending on this similarity, we modify the number of training cycles for the new dataset. A higher similarity leads to fewer training cycles and reduced training duration. This strategy allows us to cut down the overall training time by an average of 48% without sacrificing generalization capabilities.

To summarize, the key contributions of our paper are outlined below:

- Inspired by the complementary learning system in human brain, we propose a novel continual learning paradigm for image de-raining, emphasizing human-like knowledge accumulation, offering a novel perspective on enhancing model generalization.
- We propose a generalized de-raining framework (CLGID) that integrates generative replay, interleaved training, and consistency-based distillation to accumulate de-raining knowledge across multiple datasets while alleviating catastrophic forgetting.
- We introduce a similarity-based training speedup mechanism that reduces training iterations for new datasets based on their similarity to previously learned ones, expediting training without compromising generalization.

II. Related works

A. Single Image De-raining

Recent years have witnessed significant progress in image de-raining. Most traditional methods for addressing this problem employ kernels [32], low-rank approximation [33], and dictionary learning [34]. However, due to using the handcrafted features to estimate the rain model in traditional de-raining methods, they fail under complex rain conditions and produce degraded image contents. Recently, deep learning-based methods [35–40] have emerged for rain streak removal and achieved impressive restoration performance. AirNet [41] proposes a contrastive-based all-in-one restoration framework that handles diverse unknown corruptions, including de-raining, without requiring prior degradation information, and shows strong flexibility in real-world scenarios.

CLEARER [42] introduces a NAS-based multi-scale image restoration architecture that adaptively balances performance and complexity, replacing handcrafted design. MaIR [43] leverages a Mamba-based structure with continuity-preserving scanning and sequence attention, and achieves state-of-the-art results across multiple restoration tasks, including de-raining. DPCNet [44] introduces a dual-path spatial–frequency interaction network with adaptive fusion, effectively restoring rain-corrupted details across diverse conditions. DLINet [45] adopts a decoupled architecture for rain location and intensity, mitigating feature interference and improving sub-task specialization. DualCNN [46] introduces a dual-branch network for joint raindrop and rain streak removal, combining detail restoration and color enhancement with guided filtering and skip connections. PLSA [47] presents a 3DLUT-based enhancement framework with pixel-adaptive intensity modeling and saturation-aware correction, improving visibility and perceptual quality in degraded or low-light regions More deep learning-based methods can be found in [48, 49].

B. Accumulating De-raining Knowledge

While promising, most existing methods are trained on a fixed dataset, focusing on learning specific rain patterns. However, relying solely on such static training data is insufficient to cope with the complex and diverse rain distributions encountered in real-world images. When exposed to unseen rainy scenarios, these networks often experience a notable drop in performance. Therefore, it is important to enable de-raining networks to continually accumulate knowledge from increasingly diverse datasets, rather than depending entirely on a fixed one, allowing them to steadily enhance their generalization ability in practical applications. Recent studies [18, 22, 23] explore equipping de-raining networks with continual learning capabilities. For instance, PIGWM [16] develops a parameter-importance-guided weight modification method that enables de-raining networks to learn from a sequence of datasets. NR [17] explores a neural reorganization method to ensure the accumulation of de-raining knowledge and overcome catastrophic forgetting. Gu *et al.* [50] propose a memory management strategy, called associative memory, that links the current pathway with historical representations to enable incremental rain removal. However, these methods accumulate de-raining knowledge effectively only within a limited number of datasets. As more datasets are introduced, their memory capacity becomes saturated, restricting knowledge accumulation and hindering further improvement in generalization. In contrast, our CLGID, inspired by the complementary learning system, imitates the coordinated roles of rapid learning in the hippocampus and gradual integration in the neocortex to maintain memory capacity and prevent saturation. Benefiting from this brain-inspired mechanism, CLGID continually enhances generalization as more datasets arrive and achieves superior performance on unseen real-world rainy images.

Notably, recent work MiOIR [51] highlights the benefit of introducing datasets in an orderly manner so that earlier-learned datasets serve as a pre-training foundation for later ones, which could inspire more principled training orders in accumulating de-raining knowledge to reduce conflicts between new and old knowledge.

C. Continual Learning

At the methodological level, our proposed framework can be more broadly categorized as continual learning (CL) [52], whose goal is to enable a model to acquire new knowledge from a stream of data or tasks while mitigating catastrophic forgetting. Existing CL methods can be broadly divided into three categories. The first category, regularization-based methods [53], introduces constraints into the objective function to preserve prior knowledge, thereby limiting parameter or output drift. The second category [54], replay-based methods, explicitly consolidate old knowledge by either storing a small set of historical samples or using generative models to synthesize past data, which are then mixed with new data during training. The third category [55], parameter-isolation methods, allocates relatively independent parameter subsets or pathways to different tasks, reducing interference between them.

Most general CL methods are validated on high-level tasks such as image classification, where the knowledge to retain primarily consists of high-level semantic features that define decision boundaries. In contrast, image de-raining is a low-level, pixel-wise reconstruction task, where retention centers on the statistical and structural properties of the degradation process, including rain streak orientation, density, and scale. This fundamental difference in task nature leads to a different knowledge retention target: classification focuses on the separability of high-level semantics, whereas de-raining focuses on modeling the degradation mapping. As an ill-posed inverse problem, de-raining can be approximated as the additive combination of a background layer and a rainy layer. Exploiting this property, we focus memory retention on the degradation: for each previously seen dataset, we train a generator to model only the distribution of degradations, decoupled from the background. During interleaved learning and distillation, previous degradations are combined with new backgrounds, concentrating gradients on preserving degradation removal ability while avoiding interference from background shifts. Distillation is applied at the output level to directly stabilize reconstructed images rather than high-level features, making it better aligned with pixel-wise regression objectives. These specific designs precisely ground the general CL paradigm in the key variable of de-raining, i.e., degradation process.

D. Domain Generalization

Consistent with the central goal of this work, domain generalization (DG) [56] aims to learn a model from data in one or multiple related yet distinct domains so that it can perform well in unseen scenarios. Existing DG methods can be broadly categorized into four groups: (1) domain alignment [57], which aligns feature distributions across domains to reduce domain shift; (2) data augmentation [58], which diversifies training data through style perturbations or generative synthesis to broaden distributional coverage; (3) self-supervised learning [59], which employs auxiliary tasks on unlabeled data to learn domain-invariant representations; and

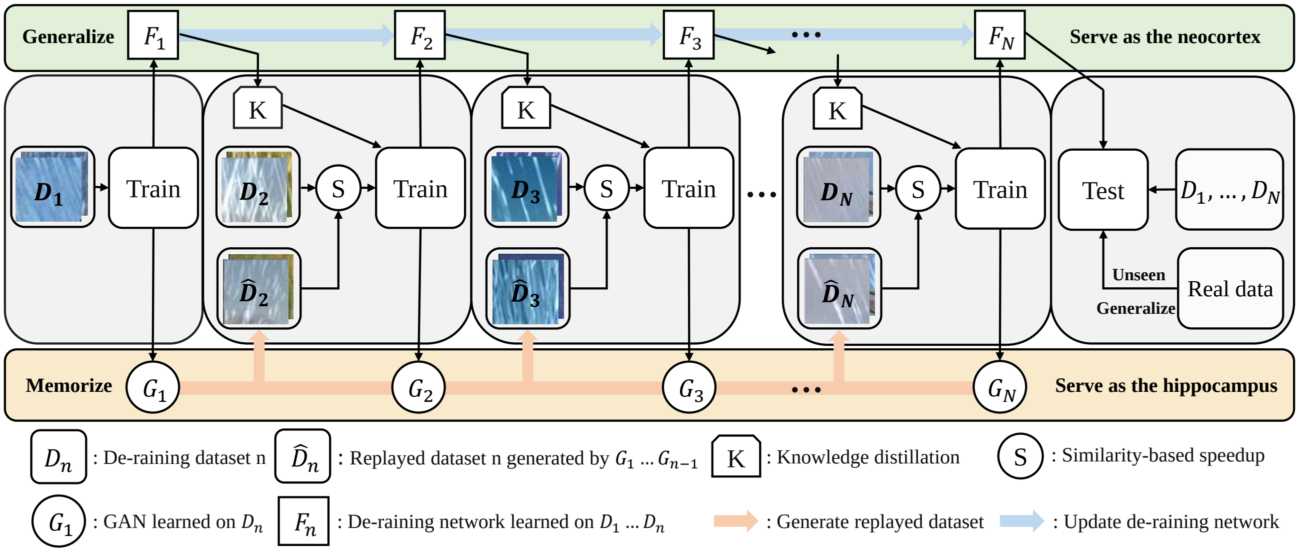

Generalize
F_1
F_2
F_3
F_N
Serve as the neocortex
K
K
K
D_1
Train
D_2
S
Train
D_3
S
Train
D_N
S
Train
Test
$D_1, \dots, D_N$
$\hat{D}_2$
$\hat{D}_3$
$\hat{D}_N$
Unseen
Generalize
Real data
Memorize
G_1
G_2
G_3
G_N
Serve as the hippocampus
D_n : De-raining dataset n
$\hat{D}_n$: Replayed dataset n generated by $G_1 \dots G_{n-1}$
K : Knowledge distillation
S : Similarity-based speedup
G_1 : GAN learned on D_n
F_n : De-raining network learned on $D_1 \dots D_n$
: Generate replayed dataset
: Update de-raining network

Fig. 3: Flowchart of the proposed CLGID framework. Given each incoming dataset D_n, a corresponding GAN (G_n) mimics the hippocampus to learn and store rain characteristics. Previously trained GANs replay past data ($\hat{D}_n$), interleaved with current data (D_n), to train the de-raining network (F_n), analogous to the neocortex. Knowledge distillation ensures consistency between current and past knowledge. A similarity-based training speedup algorithm further reduces training iterations.

(4) learning disentangled representations [60], which decouple domain-invariant components and suppresses domain-specific variations to emphasize invariant information. Unlike DG methods that enhance generalization from a static perspective, this work targets temporally varying, continual scenarios with the goal of dynamically improving the model's generalization capability. By sequentially learning from a series of de-raining datasets, the proposed approach continually accumulates de-raining knowledge, thereby expanding the coverage of rain pattern distributions and enabling the model to maintain superior performance when facing unseen real-world rainy scenes.

III. Methodology

Given a stream of de-raining datasets $\{D_i\}_{i=1}^N$, where each D_i contains M_i pairs of rainy images x_i and clean backgrounds y_i, our goal is to constantly improve the de-raining network generalization by accumulating de-raining knowledge from increasingly abundant datasets. To achieve this goal, we seek inspiration from the human brain. Humans possess the ability to constantly learn and memorize a stream of perceived events, extracting statistical structures across memorized events to acquire the generalization ability to unseen situations. The complementary learning system, comprised of the hippocampus and neocortex, significantly contributes to the aforementioned process. Inspired by the complementary learning system, we propose CLGID, a generalized image de-raining framework that imitates this mechanism to maintain effective memory capacity and improve generalization. The overall framework is shown in Fig. 3.

A. *Imitating the complementary learning system*

In the complementary learning system, the hippocampus allows for the learning and individualized storage of a stream of perceived event. Accordingly, we adopt GANs, as exemplified by VRGNet [61], to learn and store the rain streak characteristics for each incoming dataset individually, which play the role of the hippocampus. Specifically, with the arrival of a new dataset $D_{n\in[1,N]}$, a corresponding GAN G_n will be trained on D_n to learn the rain streak characteristics of D_n. The forward propagation process of learning the rain streak characteristics can be formulated as:

$$\alpha_n, \beta_n = R_i\left(x_n; W_{R_i}\right), \tag{1}$$

$$z_n \leftarrow \text{Reparameterize}(\alpha_n, \beta_n), \tag{2}$$

$$r_n = R_g\left(z_n; W_{R_g}\right), \tag{3}$$

where R_i and R_g indicate the rain inference network and rain generator in VRGNet, W_{R_i} and W_{R_g} denote the parameters of R_i and R_g, α_n and β_n are the posterior parameters (i.e., mean and variance, respectively) of latent variable z_n inferenced by R_i, r_n represents the rain streak layer of x_n generated by R_g. More training details can be found in [61]. Consequently, we have obtained the G_n, which store the rain streak characteristics of D_n.

Then, the hippocampus will repeatedly replay the memorized events in the hippocampus, interleaved with the new events, back to the neocortex. To imitate the hippocampus-to-neocortex replay and the interleaved learning, we first construct a replayed dataset $\hat{D}_n$ generated by previously learned GANs $\{G_1, G_2, \cdots, G_{n-1}\}$ that are trained on $\{D_1, D_2, \cdots, D_{n-1}\}$, respectively. Then, the replayed dataset $\hat{D}_n$ and the new dataset D_n are utilized for training the de-raining network F_n, which serves as the neocortex. Specifically, for replayed dataset $\hat{D}_n$, we determine which GAN generates each pair of images by uniformly sampling from the previously learned GANs, which can be expressed as:

$$G_r \leftarrow \{G_1, G_2, \cdots, G_{n-1}\}. \tag{4}$$

Moreover, we sample the latent variable $\hat{z}_n$ from isotropic Gaussian distribution and use the rain generator R_g of G_r to generate the rain streak layer $\hat{r}_n$ [61]:

$$\hat{z}_n \leftarrow \mathcal{N}(0, \mathbf{I}_t), \tag{5}$$

$$\hat{r}_n = R_g(\hat{z}_n; W_{R_g}), \tag{6}$$

where $\mathbf{I}_t \in \mathbb{R}^{t \times t}$ is the unit matrix. For each generated rain layer $\hat{r}_n$, we randomly select a clean background image y_n from $D_n = \{x_n^m, y_n^m\}_{m=1}^{M_n}$ to form a replayed rainy image $\hat{x}_n$ by adding $\hat{r}_n$ to y_n. Therefore, the replayed dataset $\hat{D}_n$ can be recorded as:

$$\hat{D}_n = \{y_n^m + \hat{r}_n^m, y_n^m\}_{m=1}^{M_n} \triangleq \{\hat{x}_n^m, \hat{y}_n^m\}_{m=1}^{M_n}. \tag{7}$$

In addition, throughout this iterative process of replaying hippocampal memories and interleaved learning, the neocortical activity patterns activated by the hippocampus' replayed events remain consistent with existing neocortical knowledge. To imitate this characteristic, we employ knowledge distillation with replayed data to ensure the consistency of the knowledge in the de-raining network. Specifically, we distill the de-raining knowledge from the previous obtained de-raining network F_{n-1} that trained with the arrival of D_{n-1} to the current de-raining network F_n. We send the replayed rainy images of $\hat{D}_n$ to both F_{n-1} and F_n and ensure the consistency of the knowledge by encouraging the outputs of F_{n-1} and F_n to be similar. Thus, the neocortex gradually extracts structured knowledge across events and acquires the ability to generalize to unseen situations.

To sum up, the total loss function of the proposed framework comprises the interleave loss, which includes the new loss and the replay loss, along with the consistency loss, which enables the de-raining network to constantly acquire generalization ability to unseen real-world data after training on a stream of de-raining datasets:

$$\mathcal{L}_{\text{new}} = \mathcal{L}_{\text{char}}(F_n(x_n), y_n) + \mathcal{L}_{\text{edge}}(F_n(x_n), y_n), \tag{8}$$

$$\mathcal{L}_{\text{replay}} = \mathcal{L}_{\text{char}}(F_n(\hat{x}_n), \hat{y}_n) + \mathcal{L}_{\text{edge}}(F_n(\hat{x}_n), \hat{y}_n), \tag{9}$$

$$\mathcal{L}_{\text{interleave}} = \mathcal{L}_{\text{replay}} + \mathcal{L}_{\text{new}}, \tag{10}$$

$$\mathcal{L}_{\text{consist}} = \left\| F_n\left(\hat{x}_n\right) - F_{n-1}\left(\hat{x}_n\right) \right\|_1, \tag{11}$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{interleave}} + \lambda \times \mathcal{L}_{\text{consist}}, \tag{12}$$

where the $\mathcal{L}_{\text{char}}$ and $\mathcal{L}_{\text{edge}}$ are the loss functions of the de-raining network, as exemplified by MPRNet [29], λ is the hyper-parameter for balancing the interleave loss and the consistency loss, the new loss and the replay loss are considered to be of equal significance as the size of the new dataset D_n is the same as the replayed dataset $\hat{D}_n$. The complexity analysis of the proposed CLGID framework is provided in Section I of Supplementary Material, detailing how its parameter count, FLOPs, and time cost evolve as the number of datasets increases.

B. Similarity-based training speedup algorithm

Recent neuroscience studies [30, 31] have updated the complementary learning system theory, demonstrating that the neocortex can extract structured knowledge across events faster than originally suggested if new events are highly similar to previously learned events. Inspired by these findings, we design a similarity-based training speedup algorithm to complement our CLGID framework. We calculate the rain characteristics similarity between the new and previously learned datasets and reduce the total training iterations and training time for the new dataset according to its similarity to the previously learned datasets. The greater the similarity, the fewer the total training iterations and the shorter the training time. Specifically, with the arrival of new dataset $D_n = \{x_n^m, y_n^m\}_{m=1}^{M_n}$, we utilize the replayed dataset $\hat{D}_n$ constructed for training D_n, which is composed by uniformly sampling from all previously trained GANs $\{G_1, G_2, \cdots, G_{n-1}\}$. To compute similarity, we divide $\hat{D}_n$ into subsets $\{\hat{D}_1, \hat{D}_2, \cdots, \hat{D}_{n-1}\}$ according to GAN sources and extract Histogram of Oriented Gradients (HOG) [62] from the rainy images in both $\hat{D}_{i \in \{1,\ldots,n-1\}}$ and D_n as:

$$\hat{h}_i = \text{HOG}(\hat{x}_i), \quad h_n = \text{HOG}(x_n). \tag{13}$$

The Kullback-Leibler (KL) divergence [63] is then used to compute the similarity between $\hat{D}_i$ and D_n:

$$s_i^n = D_{KL}(\hat{h}_i || h_n), \tag{14}$$

where s_i^n denotes the similarity coefficient between $\hat{D}_i$ and D_n. After calculating all similarity coefficients $\{s_1^n, s_1^n, \cdots, s_{n-1}^n\}$, we select the smallest of these values as the final similarity coefficient S_n between the new dataset D_n and the previously learned datasets:

$$S_n = \min_{i \in \{1,\ldots,n-1\}} s_i^n. \tag{15}$$

Since the value range of KL divergence is $[0, +\infty]$, we utilize a mapping function to map its value range from $[0, +\infty]$ to $[0, 1]$ for bounding its value domain:

$$\widehat{S}_n = 1 - e^{-S_n}. \tag{16}$$

Finally, we reduce the total training iterations from I_n to $\widehat{I}_n$ for the new dataset D_n based on the final similarity coefficient $\widehat{S}_n$, thus reducing the training time:

$$\widehat{I}_n = \widehat{S}_n \times I_n. \tag{17}$$

C. Improving scalability of the framework

Similarity-Based Selective GAN Training: Current framework trains one GAN for each new dataset, which leads to linearly growing training and storage cost. Inspired by the similarity-based training speedup mechanism, we extend this idea to GAN training and determine whether a new GAN should be trained for the incoming dataset $D_n = \{x_n^m, y_n^m\}_{m=1}^{M_n}$. Following the similarity computation pipeline defined in Eq. 13–16, we first compute the similarity s_i^n between D_n and each GAN-generated replayed dataset $\hat{D}_{i \in \{1,\ldots,n-1\}}$. Let $S_n = \min_{i \in \{1,\ldots,n-1\}} s_i^n$, we denote the normalized divergence between the current dataset D_n and all replayed datasets $\{\hat{D}_i\}_{i=1}^{n-1}$ as $\hat{S}_n = 1 - e^{-s_n} \in [0, 1]$. A higher value of $\hat{S}_n$ implies greater dissimilarity.

To improve the scalability of GAN training, we train a new GAN G_n only when the normalized similarity score $\hat{S}_n$ exceeds a predefined threshold $\hat{T} \in (0, 1)$:

$$\text{Train } G_n \quad \text{if } \hat{S}_n > \hat{T}. \tag{18}$$

Let the binary indicator be defined as:

$$\delta_n = \begin{cases} 1, & \text{if } \hat{S}_n > \hat{T}, \\ 0, & \text{otherwise}, \end{cases} \tag{19}$$

then the total number of trained GANs after N stages is:

$$|\mathcal{G}_N| = \sum_{i=1}^{N} \delta_i. \tag{20}$$

With the increasing number of trained GANs, the diversity of captured rain patterns becomes more comprehensive, lowering the probability of requiring additional GAN training. As a result, both GAN training and storage costs grow sub-linearly, thus improving the overall scalability of the framework.

GAN-replayed Data Reuse: In current framework, each new dataset D_n requires generating a replayed dataset $\hat{D}_i$ of the same size by uniformly sampling from all previously trained GANs $\{G_i\}_{i=1}^{n-1}$. This leads to M_n GAN forward passes at every stage, and a total replay cost of $\mathcal{O}(MN)$, which scales linearly with the number of datasets and poses a practical limitation.

To address this scalability bottleneck, we propose a replay data reuse mechanism. The core idea is to reuse replayed samples generated in previous stages and only perform incremental generation if necessary. Specifically, at stage n, the replayed dataset $\hat{D}_n$ is constructed by generating $M_n/(n-1)$ samples using each of the previously trained GANs $\{G_1, \ldots, G_{n-1}\}$, where M_n is the size of the current dataset D_n. We denote $r_{i,n} = M_n/(n-1)$ as the required number of replayed samples from G_i at stage n. Meanwhile, let $c_{i,n-1} = M_{n-1}/(n-2)$ be the number of cached samples generated by G_i during the previous stage $n-1$. Then, for each $i = \{1, \ldots, n-2\}$, we reuse the cached samples and generate only the difference:

$$\Delta_{i,n} = \max(0, r_{i,n} - c_{i,n-1}).$$

For the newly introduced GAN G_{n-1}, which was not involved in $\hat{D}_{n-1}$, we generate all its required samples:

$$\Delta_{n-1,n} = r_{n-1,n} = \frac{M_n}{n-1}.$$

To analyze the total replay cost up to stage N, we consider the total number of additional forward passes across all stages:

$$\begin{aligned} C_N &= \sum_{n=2}^{N} \left(\sum_{i=1}^{n-2} \Delta_{i,n} + \Delta_{n-1,n} \right) \\ &= \sum_{n=2}^{N} (n-2) \cdot \max\left(0, \frac{M_n}{n-1} - \frac{M_{n-1}}{n-2}\right) + \frac{M_n}{n-1}. \end{aligned} \tag{21}$$

As shown by the mathematical proof in Section II of Supplementary Material, we can conclude that:

$$C_N = \mathcal{O}(M \log N), \tag{22}$$

demonstrating that the total replay cost grows logarithmically with the number of datasets, thereby enhancing scalability over the original linear design.

IV. Experiments

A. Experimental Settings

Datasets. We use six de-raining datasets for training: Rain100H [14], Rain100L [14], Rain1400 [19], and the light-, medium-, and heavy-density subsets of Rain1200 [20] (denoted as Rain1200L, Rain1200M, and Rain1200H). Rain100H and Rain100L each contain 1,800 training and 200 testing images, while Rain1400 provides 12,600/1,400 images with fourteen types of streak orientations and magnitudes. Each Rain1200 subset includes 4,000/400 images, representing different rain-density levels. For memory performance evaluation, after sequentially training on the dataset stream, each method is tested on the corresponding test set of every dataset in the stream. For generalization performance, following [17], we adopt SPA-data [21] and Real-Internet [21], which contain real-world rainy images unseen during training, thus introducing both cross-dataset and synthetic-to-real domain gaps. SPA-data offers 29,500 high-quality rain/clean pairs from 170 real rain videos (28,500/1,000 split), and we use its test set for evaluation. Real-Internet contains 146 real rainy images without ground-truth labels, collected from the Internet.

Comparison methods. We conduct comparative experiments on memory and generalization ability, utilizing a baseline (denoted by "SF"), along with three related state-of-the-art methods: PIGWM [16], NR [17], CID [22], DPL [18], and CLAIO [23]. SF entails the sequential fine-tuning of the de-raining network on each new incoming dataset. Additionally, we introduce individual training (denoted by "Individual"), which trains and tests on each single dataset within a stream of datasets, as a reference to evaluate the memory performance of several methods.

Implementation Details. Memory and generalization performance are evaluated in terms of Peak Signal-to-Noise Ratio (PSNR) [64], and Structural Similarity (SSIM) [65] metrics. We compute PSNR and SSIM metrics over RGB channels for color images. Our approach is implemented in PyTorch using NVIDIA 3090 GPUs. To ensure a fair comparison, we set the patch size of all methods to 64, including baseline, SOTA methods, and our approach. We conduct experiments on three representative de-raining networks: MFDNet [13], Restormer [28], and MPRNet [29]. The training settings of the de-raining networks remain consistent with their publicly released code, including batch size, training epochs, iterations, optimizer, scheduler, etc. Note that DPL is designed for transformer-based de-raining networks; therefore, we only evaluate DPL on MFDNet and Restormer. The hyper-parameter of our framework, λ, which balances the interleave loss and the consistency loss, is set to 1. The $\hat{T}$ in similarity-based selective GAN training is set to 0.4.

TABLE I: Qualitative comparison of memory and generalization performance after training on a stream of four datasets in four distinct sequences. All comparison methods using MFDNet [13] as the de-raining network. CLGID† represents the accelerated training version of CLGID, using our proposed similarity-based training speedup algorithm. Individual signifies training and testing on each dataset individually, providing a reference for evaluating the memory performance. We evaluate generalization on SPA-data, ***which has never been seen during training***. We highlight the best results using **such** formatting.

Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1400-1200M-100H-100L	Individual	31.79	0.920	32.24	0.920	27.70	0.886	36.16	0.978	31.97	0.926	–	–
	SF	28.67	0.878	27.00	0.833	25.54	0.840	36.34	0.977	29.39	0.882	33.59	0.941
	PIGWM	29.18	0.893	27.74	0.963	26.38	0.863	33.85	0.965	29.29	0.921	33.94	0.947
	NR	30.21	0.908	27.95	0.874	18.39	0.615	31.74	0.939	27.07	0.834	33.70	0.947
	CID	29.01	0.880	29.67	0.882	24.91	0.825	34.03	0.960	29.41	0.887	33.03	0.939
	DPL	29.64	0.876	28.21	0.875	25.57	0.844	34.61	0.962	29.51	0.889	32.54	0.931
	CLAIO	30.01	0.901	30.53	0.893	27.36	0.882	34.78	0.961	30.67	0.909	33.98	0.946
	CLGID	31.26	0.915	31.65	0.905	28.07	0.891	36.77	0.979	**31.94**	**0.923**	**34.12**	**0.948**
	CLGID†	31.09	0.912	30.65	0.893	27.94	0.889	36.65	0.979	31.58	0.918	34.06	0.947
Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1400-100L-1200M-100H	Individual	31.79	0.920	36.16	0.978	32.24	0.920	27.70	0.886	31.97	0.926	–	–
	SF	28.61	0.879	36.19	0.976	26.85	0.837	28.57	0.900	30.06	0.898	33.70	0.943
	PIGWM	29.59	0.899	33.35	0.964	29.22	0.884	26.51	0.865	29.67	0.903	33.48	0.948
	NR	30.23	0.904	31.22	0.938	28.64	0.871	19.96	0.691	27.51	0.851	32.20	0.940
	CID	28.57	0.878	32.43	0.940	29.01	0.880	27.58	0.884	29.40	0.896	33.51	0.948
	DPL	29.34	0.886	33.27	0.961	28.07	0.869	24.47	0.839	28.79	0.889	33.89	0.948
	CLAIO	30.82	0.903	35.01	0.969	28.30	0.871	27.41	0.884	30.39	0.907	34.10	0.949
	CLGID	31.24	0.916	36.21	0.978	31.74	0.908	27.68	0.887	**31.72**	**0.922**	**34.42**	**0.952**
	CLGID†	31.06	0.906	36.27	0.978	30.05	0.883	27.66	0.884	31.26	0.913	34.24	0.949
Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
100L-100H-1400-1200M	Individual	36.16	0.978	27.70	0.886	31.79	0.920	32.24	0.920	31.97	0.926	–	–
	SF	24.62	0.819	13.02	0.369	28.98	0.891	32.40	0.922	24.76	0.750	29.09	0.917
	PIGWM	26.08	0.872	14.02	0.435	28.40	0.891	30.85	0.895	24.84	0.773	28.16	0.908
	NR	31.98	0.951	17.13	0.564	29.81	0.898	28.01	0.869	26.73	0.821	32.55	0.933
	CID	32.01	0.955	21.62	0.770	28.10	0.888	29.67	0.871	27.85	0.871	33.01	0.941
	DPL	31.71	0.953	20.70	0.752	29.95	0.902	29.35	0.875	27.93	0.871	32.99	0.940
	CLAIO	33.68	0.969	24.31	0.820	29.89	0.901	31.62	0.912	29.88	0.901	33.56	0.944
	CLGID	35.62	0.975	26.09	0.847	31.36	0.920	32.00	0.914	**31.27**	**0.914**	**34.20**	**0.951**
	CLGID†	34.82	0.972	25.69	0.823	31.18	0.920	31.93	0.916	30.91	0.908	34.09	0.949
Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
100H-100L-1400-1200M	Individual	27.70	0.886	36.16	0.978	31.79	0.920	32.24	0.920	31.97	0.926	–	–
	SF	12.98	0.365	24.93	0.826	28.83	0.887	32.37	0.922	24.78	0.750	29.12	0.917
	PIGWM	14.58	0.461	26.14	0.869	29.08	0.901	30.76	0.894	25.14	0.781	28.16	0.908
	NR	23.66	0.802	31.73	0.945	29.76	0.898	29.16	0.879	28.58	0.881	32.55	0.933
	CID	24.98	0.813	32.11	0.942	29.54	0.891	29.31	0.878	28.99	0.881	33.01	0.941
	DPL	20.84	0.736	32.17	0.941	28.32	0.872	30.69	0.883	28.01	0.858	32.63	0.940
	CLAIO	24.02	0.808	35.00	0.971	30.28	0.909	31.25	0.910	30.14	0.900	33.50	0.942
	CLGID	26.74	0.851	35.54	0.974	31.11	0.919	31.90	0.914	**31.32**	**0.915**	**33.96**	**0.949**
	CLGID†	26.29	0.832	35.18	0.971	31.10	0.918	31.78	0.911	31.09	0.908	33.82	0.948

B. Results on Benchmark Datasets

To validate the efficacy of our approach, we conduct twelve experiments using three representative de-raining networks, i.e., MFDNet [13], Restormer [28], and MPRNet [29], on four representative training sequences composed of four de-raining datasets. These sequences are deliberately designed to reflect variations in rain type diversity, intensity, and dataset complexity, enabling a systematic evaluation of the proposed method's robustness to training order in continual learning scenarios. After training, we evaluate both memory performance on the seen datasets and generalization performance on the unseen real-world SPA-data [21]. As shown in Tab. I, II, and III, sequential fine-tuning (SF) suffers from severe forgetting, leading to poor generalization. Compared with state-of-the-art methods including PIGWM [16], NR [17], CID [22], DPL [18], and CLAIO [23], our CLGID achieves substantial improvements in memory retention and matches the performance of the Individual baseline, which represents an upper bound by training separately on each dataset. Regarding generalization, CLGID consistently outperforms all competitors across all settings. Furthermore, we analyze the variation in generalization performance as more datasets are introduced, as illustrated in Fig. 4, 5, and 6. While other methods show limited or unstable improvement due to memory saturation, our method exhibits consistent gains in generalization across all experiments, demonstrating its strong knowledge accumulation ability and robustness to variations in training sequences. We also provide qualitative comparisons between CLGID and

TABLE II: Qualitative comparison of memory performance after training on a stream of four datasets in four distinct sequences. All comparison methods utilize Restormer [28] as the de-raining network.

Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1400-1200M-100H-100L	Individual	32.01	0.929	32.80	0.925	29.87	0.914	38.33	0.985	33.25	0.938	–	–
	SF	26.63	0.842	21.79	0.707	21.94	0.679	38.52	0.986	27.22	0.804	32.87	0.931
	PIGWM	27.80	0.875	23.61	0.781	25.89	0.830	36.30	0.977	28.40	0.866	33.10	0.938
	NR	31.62	0.924	29.73	0.868	18.41	0.627	29.17	0.911	27.23	0.833	33.39	0.941
	CID	28.79	0.886	30.25	0.898	25.17	0.824	32.46	0.940	29.17	0.887	33.30	0.941
	DPL	28.32	0.888	28.48	0.849	23.93	0.796	35.95	0.967	29.17	0.875	33.87	0.945
	CLAIO	30.01	0.911	31.87	0.901	27.53	0.865	35.21	0.961	31.16	0.910	33.91	0.944
	CLGID	31.74	0.923	32.18	0.912	28.59	0.889	37.37	0.982	**32.47**	**0.927**	34.03	**0.948**
	CLGID†	31.64	0.922	32.05	0.908	28.34	0.886	37.14	0.981	32.29	0.924	**34.05**	0.947
Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1400-100L-1200M-100H	Individual	32.01	0.929	38.33	0.985	32.80	0.925	29.87	0.914	33.25	0.938	–	–
	SF	28.72	0.880	38.15	0.983	27.13	0.841	30.22	0.920	31.06	0.906	32.60	0.929
	PIGWM	29.41	0.894	34.91	0.970	27.54	0.859	28.05	0.877	29.98	0.900	33.59	0.942
	NR	31.59	0.924	31.78	0.944	30.27	0.876	19.70	0.676	28.34	0.855	33.35	0.945
	CID	25.67	0.834	30.44	0.935	25.51	0.825	25.90	0.821	26.88	0.854	33.03	0.939
	DPL	23.20	0.770	25.10	0.834	20.67	0.683	11.70	0.387	20.17	0.669	33.26	0.940
	CLAIO	30.02	0.910	35.60	0.971	30.91	0.900	27.96	0.860	31.12	0.910	34.01	0.944
	CLGID	31.62	0.923	37.13	0.981	32.19	0.915	29.17	0.902	32.53	**0.930**	**34.34**	**0.952**
	CLGID†	31.74	0.924	37.03	0.981	32.30	0.916	29.14	0.900	**32.55**	**0.930**	34.10	0.947
Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
100L-100H-1400-1200M	Individual	38.33	0.985	29.87	0.914	32.01	0.929	32.80	0.925	33.25	0.938	–	–
	SF	24.49	0.824	12.87	0.366	28.80	0.890	32.78	0.925	24.74	0.751	29.67	0.919
	PIGWM	26.16	0.888	14.46	0.468	29.03	0.899	31.66	0.904	25.33	0.790	30.96	0.930
	NR	36.39	0.977	20.68	0.705	28.73	0.886	27.05	0.852	28.21	0.855	33.32	0.941
	CID	34.41	0.951	26.77	0.829	26.51	0.815	24.63	0.809	28.08	0.851	33.11	0.937
	DPL	32.20	0.922	20.86	0.717	29.14	0.896	28.55	0.869	27.69	0.851	33.49	0.942
	CLAIO	36.03	0.979	27.86	0.830	28.53	0.884	31.91	0.901	31.08	0.899	33.50	0.939
	CLGID	37.79	0.983	29.23	0.903	31.50	0.921	32.12	0.912	**32.66**	**0.930**	**34.24**	**0.945**
	CLGID†	37.08	0.980	28.13	0.885	31.33	0.919	32.01	0.911	32.14	0.924	34.13	0.941
Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
100H-100L-1400-1200M	Individual	29.87	0.914	38.33	0.985	32.01	0.929	32.80	0.925	33.25	0.938	–	–
	SF	12.92	0.366	24.64	0.825	28.93	0.887	32.83	0.925	24.83	0.751	30.07	0.930
	PIGWM	16.96	0.597	25.90	0.859	29.18	0.908	31.64	0.906	25.92	0.818	29.66	0.908
	NR	27.06	0.864	36.26	0.976	28.90	0.883	27.54	0.840	29.94	0.891	32.77	0.936
	CID	24.02	0.838	33.96	0.914	28.13	0.865	26.27	0.831	28.10	0.862	33.01	0.939
	DPL	24.95	0.845	32.58	0.924	28.54	0.876	28.85	0.857	28.73	0.876	32.96	0.938
	CLAIO	25.77	0.851	35.98	0.959	30.03	0.891	29.43	0.894	30.30	0.899	32.03	0.932
	CLGID	27.85	0.879	37.00	0.979	31.24	0.918	31.89	0.910	32.00	0.922	**33.81**	**0.943**
	CLGID†	28.49	0.891	37.13	0.980	31.38	0.920	31.98	0.912	**32.25**	**0.926**	33.45	0.942

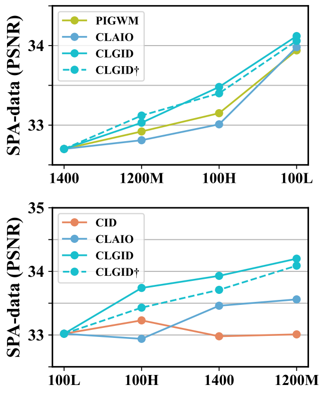


Fig. 4: Generalization variance on unseen SPA-data during training on four dataset streams across four sequences using MFDNet [13], showing the top four methods.

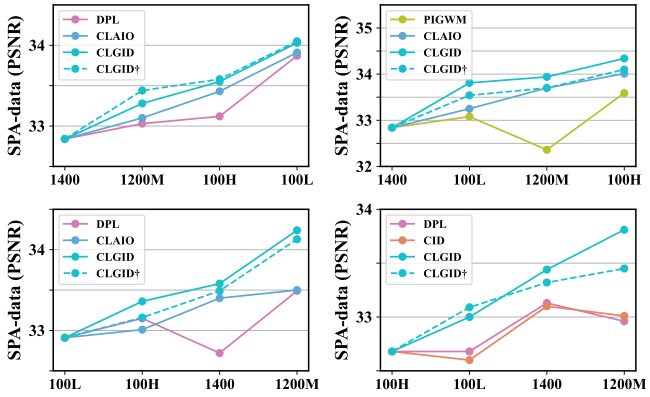


Fig. 5: Generalization variance on unseen SPA-data during training on four dataset streams across four sequences using Restormer [28], showing the top four methods.

TABLE III: Qualitative comparison of memory performance after training on a stream of four datasets in four distinct sequences. All comparison methods utilize MPRNet [29] as the de-raining network.

Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1400-1200M-100H-100L	Individual	31.63	0.922	31.93	0.906	27.28	0.885	36.35	0.979	31.80	0.923	–	–
	SF	26.66	0.845	21.75	0.711	19.57	0.623	37.69	0.983	26.42	0.791	33.05	0.936
	PIGWM	27.23	0.866	22.83	0.759	19.34	0.660	35.88	0.975	26.32	0.815	33.24	0.943
	NR	30.20	0.905	26.77	0.845	23.48	0.804	34.64	0.804	28.77	0.840	33.33	0.944
	CID	28.05	0.870	26.45	0.841	22.77	0.781	35.71	0.912	28.25	0.851	33.10	0.938
	CLAIO	29.67	0.881	29.65	0.880	26.21	0.834	36.50	0.977	30.51	0.893	33.35	0.944
	CLGID	30.83	0.912	31.27	0.900	27.46	0.869	37.37	0.981	**31.73**	**0.916**	**33.56**	**0.945**
	CLGID†	30.46	0.909	31.08	0.895	26.59	0.855	37.01	0.980	31.29	0.910	33.41	0.944
Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
1400-100L-1200M-100H	Individual	31.63	0.922	36.35	0.979	31.93	0.906	27.28	0.885	31.80	0.923	–	–
	SF	28.33	0.876	35.98	0.974	26.76	0.844	29.12	0.898	30.05	0.898	32.89	0.939
	PIGWM	29.30	0.896	33.92	0.962	28.20	0.876	27.06	0.859	29.62	0.898	33.87	0.948
	NR	30.27	0.906	33.46	0.960	28.02	0.872	23.52	0.817	28.82	0.889	33.62	0.949
	CID	28.31	0.887	33.57	0.959	27.03	0.860	25.41	0.830	28.58	0.884	32.77	0.935
	CLAIO	30.07	0.901	36.17	0.976	29.11	0.881	27.29	0.865	30.66	0.906	33.53	0.945
	CLGID	30.76	0.911	37.33	0.980	31.23	0.896	28.99	0.895	**32.08**	**0.921**	**34.26**	**0.951**
	CLGID†	30.18	0.904	37.22	0.979	30.76	0.891	28.47	0.884	31.66	0.915	34.09	0.949
Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
100L-100H-1400-1200M	Individual	36.35	0.979	27.28	0.885	31.63	0.922	31.93	0.906	31.80	0.923	–	–
	SF	24.17	0.820	12.94	0.368	28.46	0.885	32.32	0.915	24.47	0.747	27.60	0.905
	PIGWM	26.38	0.873	14.06	0.413	27.63	0.878	29.34	0.873	24.35	0.759	31.63	0.935
	NR	31.46	0.941	16.02	0.488	29.39	0.895	27.89	0.869	26.19	0.798	31.87	0.938
	CID	32.62	0.945	20.03	0.613	29.12	0.890	28.66	0.873	27.61	0.830	32.00	0.940
	CLAIO	33.41	0.951	24.91	0.770	29.10	0.888	30.10	0.889	29.38	0.875	33.90	0.948
	CLGID	35.51	0.973	27.14	0.865	30.78	0.913	31.97	0.911	**31.26**	**0.916**	**34.39**	**0.953**
	CLGID†	34.58	0.967	25.88	0.834	29.26	0.901	31.56	0.904	30.32	0.902	34.13	0.952
Training Sequence	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
100H-100L-1400-1200M	Individual	27.28	0.885	36.35	0.979	31.63	0.922	31.93	0.906	31.80	0.923	–	–
	SF	12.94	0.365	24.14	0.817	28.30	0.878	32.40	0.917	24.45	0.744	28.40	0.914
	PIGWM	13.29	0.393	25.17	0.849	26.83	0.863	28.37	0.859	23.42	0.741	32.37	0.937
	NR	15.17	0.453	30.46	0.923	28.62	0.880	27.30	0.852	25.39	0.777	33.05	0.946
	CID	20.63	0.650	30.30	0.920	29.21	0.891	28.00	0.848	27.04	0.827	33.65	0.946
	CLAIO	25.02	0.814	32.92	0.959	28.79	0.883	31.20	0.900	29.48	0.889	34.01	0.948
	CLGID	26.07	0.841	35.41	0.973	30.73	0.913	31.94	0.910	**31.04**	**0.909**	**34.60**	**0.956**
	CLGID†	25.98	0.843	34.98	0.969	30.14	0.911	31.70	0.908	30.70	0.908	34.22	0.951

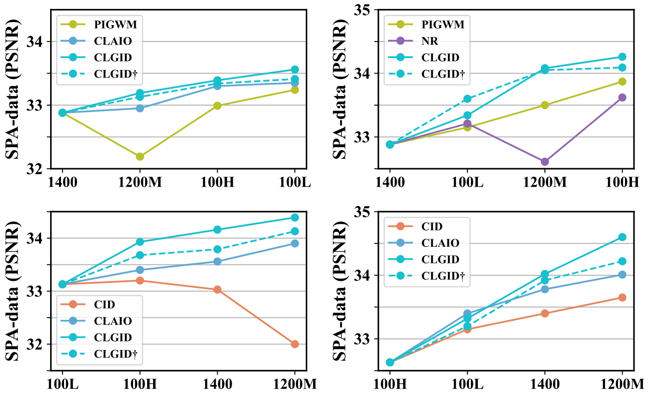


Fig. 6: Generalization variance on unseen SPA-data during training on four dataset streams across four sequences using MPRNet [29], showing the top four methods.

competing methods on SPA-data [21] and Real-Internet [21] in Section III of the Supplementary Material.

C. Extension to More Datasets

To further evaluate the memory retention and generalization capability of CLGID on more datasets, we extend the training sequence to include six datasets: 1400-1200M-100H-100L-1200L-1200H. The corresponding results on memory and generalization performance after training, as well as the generalization trend during training, are presented in Tab. IV and Fig. 7. CLGID consistently outperforms existing methods in both metrics, demonstrating strong resistance to catastrophic forgetting. Compared with its performance under the four-dataset setting (1400-1200M-100H-100L), CLGID not only preserves the knowledge of previously seen datasets but also achieves further improvements in generalization to unseen real-world images. In contrast, competing methods suffer significant drops in generalization performance as more datasets

TABLE IV: Qualitative comparison of memory performance after training on a stream of six datasets in 1400-1200M-100H-100L-1200L-1200H sequence. CLGID† represents the accelerated training version of CLGID, using our proposed similarity-based training speedup algorithm. Individual signifies training and testing on each dataset individually, providing a reference for evaluating the memory performance. We evaluate generalization on SPA-data, ***which has never been seen during training***. We highlight the best results using **such** formatting.

Network	Methods	Rain1400		Rain1200M		Rain100H		Rain100L		Rain1200L		Rain1200H		Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
MFDNet	Individual	31.79	0.920	32.24	0.920	27.70	0.886	36.16	0.978	36.22	0.958	29.91	0.889	32.34	0.925	–	–
	SF	26.88	0.877	30.78	0.914	13.73	0.384	23.16	0.804	27.16	0.900	30.83	0.897	25.42	0.796	31.76	0.929
	PIGWM	29.09	0.899	30.38	0.900	14.72	0.446	25.59	0.850	31.34	0.921	29.83	0.878	26.83	0.816	32.39	0.940
	NR	31.16	0.917	30.34	0.899	15.41	0.493	29.53	0.916	34.87	0.948	26.30	0.827	27.94	0.833	32.59	0.943
	CID	29.67	0.900	28.61	0.876	23.01	0.795	31.73	0.930	30.76	0.918	25.21	0.801	28.17	0.870	32.51	0.941
	DPL	30.51	0.886	29.74	0.885	22.33	0.782	28.88	0.902	30.30	0.903	26.77	0.827	28.09	0.864	32.08	0.938
	CLAIO	30.01	0.901	30.89	0.905	24.85	0.811	33.55	0.940	33.79	0.942	28.71	0.854	30.30	0.892	34.00	0.946
	CLGID	30.73	0.908	31.07	0.913	27.30	0.865	36.49	0.978	35.97	0.954	30.40	0.889	**31.99**	**0.918**	**34.36**	**0.952**
	CLGID†	30.99	0.912	31.51	0.916	26.60	0.842	36.20	0.976	36.03	0.957	30.37	0.887	31.95	0.915	34.23	0.950
Restormer	Individual	32.01	0.929	32.80	0.925	29.87	0.914	38.33	0.985	36.54	0.960	30.95	0.898	33.42	0.935	–	–
	SF	27.38	0.878	31.18	0.918	13.42	0.374	23.67	0.812	27.84	0.912	31.17	0.900	25.78	0.799	30.14	0.915
	PIGWM	27.98	0.886	30.20	0.891	14.45	0.434	25.00	0.837	30.53	0.899	30.19	0.880	26.39	0.805	32.24	0.929
	NR	31.27	0.919	30.55	0.888	13.95	0.435	27.63	0.888	35.22	0.949	25.97	0.825	27.43	0.817	33.06	0.941
	CID	30.09	0.908	28.77	0.860	16.32	0.600	30.24	0.922	33.71	0.930	25.42	0.820	27.43	0.840	33.50	0.943
	DPL	29.95	0.907	30.43	0.882	15.73	0.554	25.85	0.867	33.70	0.933	25.56	0.826	26.87	0.828	32.28	0.930
	CLAIO	31.34	0.920	30.21	0.879	26.27	0.853	35.90	0.960	36.01	0.952	27.98	0.844	31.29	0.901	33.80	0.948
	CLGID	31.70	0.923	32.02	0.915	28.26	0.884	37.26	0.981	36.20	0.957	30.51	0.885	**32.66**	**0.924**	34.34	**0.951**
	CLGID†	31.63	0.922	31.93	0.913	28.18	0.882	37.16	0.981	36.09	0.956	30.49	0.888	32.58	**0.924**	**34.35**	0.950
MPRNet	Individual	31.63	0.922	31.93	0.906	27.28	0.885	36.35	0.979	35.02	0.942	28.08	0.856	31.72	0.915	–	–
	SF	26.24	0.867	30.54	0.907	13.59	0.379	22.48	0.783	25.91	0.877	30.57	0.888	24.89	0.784	28.54	0.897
	PIGWM	29.04	0.904	30.60	0.899	16.14	0.526	26.85	0.884	32.61	0.936	29.66	0.871	27.48	0.837	32.13	0.937
	NR	30.68	0.909	30.02	0.896	14.34	0.457	29.30	0.916	34.62	0.945	26.36	0.844	27.55	0.828	32.34	0.939
	CID	28.77	0.881	30.24	0.880	19.87	0.619	32.54	0.921	32.09	0.931	26.82	0.845	28.39	0.846	33.85	0.947
	CLAIO	30.02	0.905	30.15	0.880	25.57	0.823	34.03	0.967	33.81	0.939	28.71	0.860	30.38	0.896	34.01	0.950
	CLGID	30.68	0.907	31.15	0.906	26.91	0.858	35.69	0.974	35.23	0.944	30.22	0.882	**31.65**	**0.912**	34.30	0.953
	CLGID†	29.76	0.894	31.17	0.907	26.47	0.849	35.22	0.972	35.15	0.942	30.19	0.881	31.33	0.908	**34.34**	**0.955**

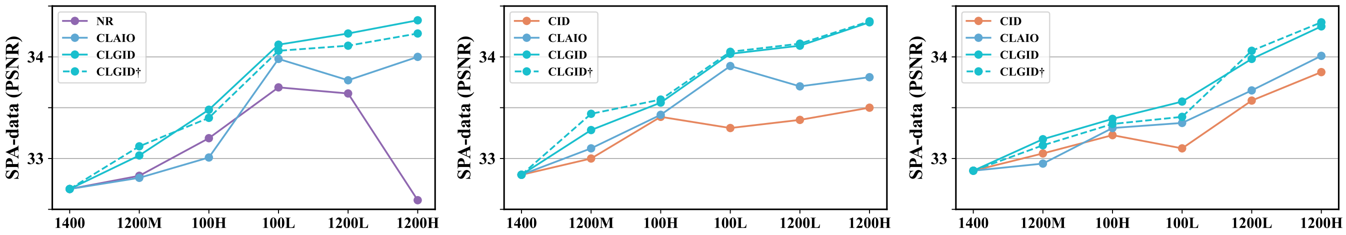


Fig. 7: Generalization performance variance on unseen SPA-data during training on a stream of six datasets. We showcase the top four methods. The three charts from left to right represent using MFDNet [13], Restormer [28], and MPRNet [29].

TABLE V: Memory and generalization performance on two alternative six-dataset sequences using MPRNet [29].

Training Sequence	Methods	Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM
1400-100L-1200M-100H-1200H-1200L	Individual	31.72	0.915	–	–
	SF	28.16	0.851	30.01	0.923
	PIGWM	29.01	0.875	33.24	0.940
	NR	27.94	0.834	33.60	0.947
	CLGID	**31.92**	**0.919**	**34.29**	**0.955**
	CLGID†	31.51	0.913	34.21	0.951
100L-100H-1400-1200M-1200L-1200H	Individual	31.72	0.915	–	–
	SF	22.10	0.705	28.12	0.907
	PIGWM	24.51	0.778	31.11	0.920
	NR	26.48	0.813	30.23	0.914
	CLGID	**31.12**	**0.914**	**34.45**	**0.957**
	CLGID†	30.60	0.911	34.33	0.955

are introduced, highlighting their limited memory capacity and inability to accumulate knowledge effectively.

To provide a comprehensive evaluation, we select one representative six-dataset sequence for the main analysis. This sequence integrates diverse data sources and rain patterns, introducing substantial distribution shifts that serve as a strong testbed for evaluating continual knowledge accumulation. While exhaustive enumeration of all possible permutations is infeasible, we report results on two alternative six-dataset sequences using MPRNet: 1400-100L-1200M-100H-1200H-1200L and 100L-100H-1400-1200M-1200L-1200H, as shown in Table V. Our method maintains superior performance across these variations, reinforcing the robustness of our framework.

D. Ablation Study

Validation on the training speedup algorithm. CLGID† represents the version of CLGID that utilizes the proposed similarity-based training speedup algorithm. Fig. 8 showcase the ratio of total training iterations of CLGID† compared

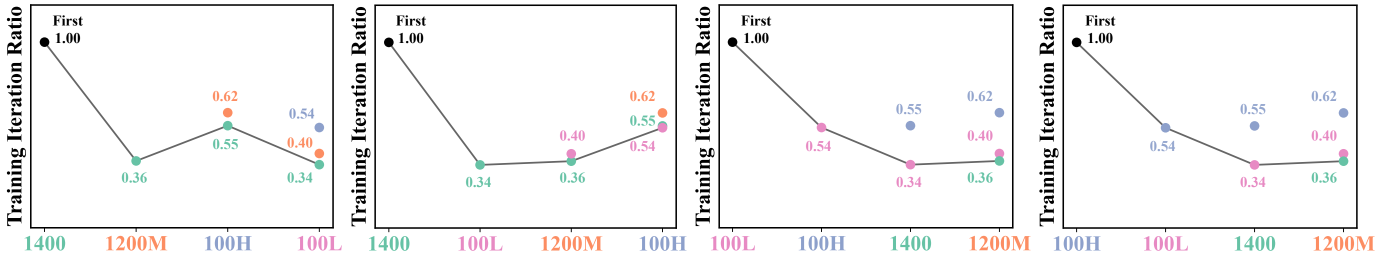
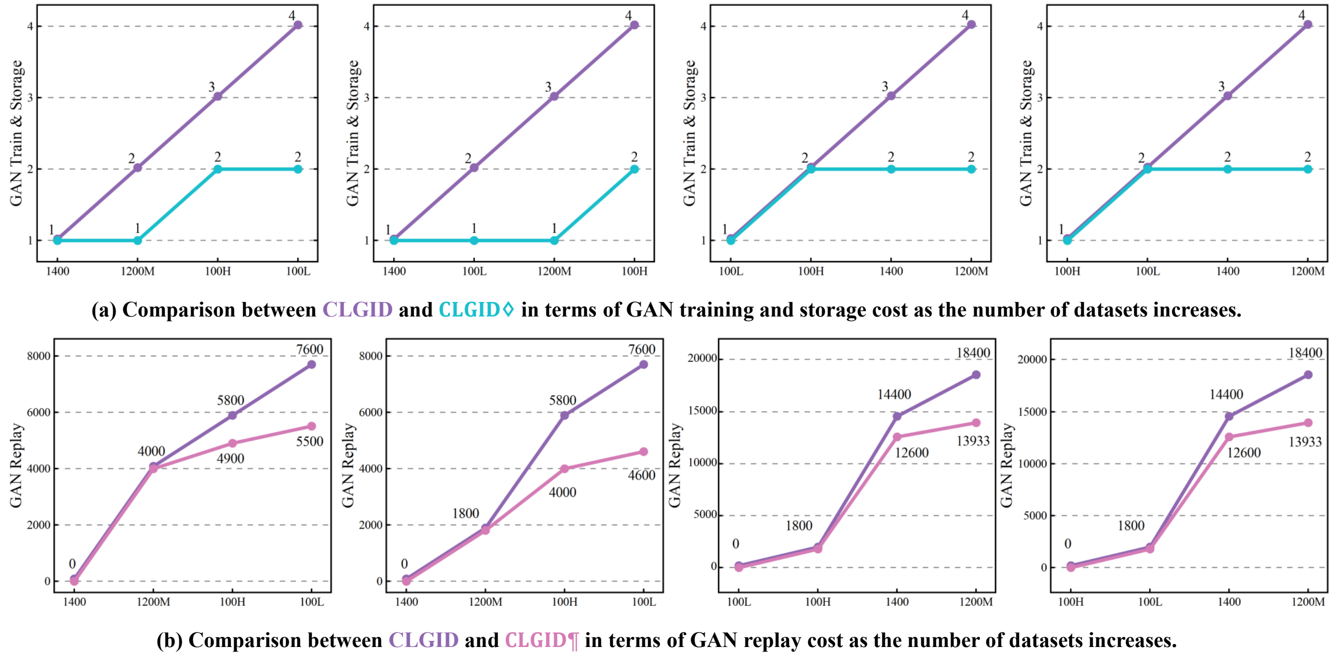


Fig. 8: Training iteration ratios of CLGID† compared with CLGID training on a stream of four datasets across four sequences. Each data point in the plots represents the similarity calculated between the current dataset and the previously learned dataset with the same color.



(a) Comparison between CLGID and CLGID◊ in terms of GAN training and storage cost as the number of datasets increases.

(b) Comparison between CLGID and CLGID¶ in terms of GAN replay cost as the number of datasets increases.

Fig. 9: Comparison of GAN training, storage, and replay costs between CLGID and its variants, CLGID$^\Diamond$ and CLGID¶, on a four-dataset stream across four sequences. The y-axis represents the number of times.

TABLE VI: Memory and generalization performance of CLGID and its variants CLGID$^\Diamond$ and CLGID¶ across four four-dataset sequences using MPRNet.

Training Sequence	Methods	Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM
1400-1200M-100H-100L	CLGID	31.73	0.916	33.56	0.945
	CLGID$^\Diamond$	31.11	0.908	33.40	0.943
	CLGID¶	31.70	0.915	33.50	0.945
1400-100L-1200M-100H	CLGID	32.08	0.921	34.26	0.951
	CLGID$^\Diamond$	31.44	0.909	34.03	0.948
	CLGID¶	32.08	0.920	34.27	0.951
100L-100H-1400-1200M	CLGID	31.26	0.916	34.39	0.953
	CLGID$^\Diamond$	30.20	0.900	34.21	0.951
	CLGID¶	31.25	0.917	34.36	0.953
100H-100L-1400-1200M	CLGID	31.04	0.909	34.60	0.956
	CLGID$^\Diamond$	30.49	0.905	34.20	0.952
	CLGID¶	31.02	0.908	34.61	0.955

with CLGID. Training on a stream of four datasets in four different sequences, CLGID† achieves an average reduction of 44% in total training iterations and 42% in total training time. The varying reductions in total training iterations and training time are attributable to the time required to calculate similarity. Furthermore, as shown in Tab. I-III and Fig. 4-6, we can observe that CLGID† achieves comparable memory and generalization performance compared to CLGID. The above observations underscore the algorithm's efficacy to shorten the training time without compromising the de-raining network's memory generalization ability, resulting in low training costs.

Validation of GAN scalability. CLGID$^\Diamond$ and CLGID¶ denote variants of the proposed CLGID framework that incorporate the similarity-based selective GAN training and GAN-replayed data reuse mechanisms, respectively. We validate their effectiveness using MPRNet on four distinct four-dataset sequences. The experimental results are presented in Fig. 9 and Tab. VI. It is evident that both CLGID$^\Diamond$ and CLGID¶ significantly enhance the scalability of the framework compared with the original CLGID. Specifically, as the number of datasets

TABLE VII: Analysis of the efficacy of each CLGID component in training on the 1400-100L-1200M-100H sequence using MPRNet [29].

	w/o both	w/o $\mathcal{L}_{\text{replay}}$	w/o $\mathcal{L}_{\text{consist}}$	w both
Rain1400	28.33 / 0.876	29.64 / 0.891	29.87 / 0.900	30.76 / 0.911
Rain100L	35.98 / 0.974	36.88 / 0.974	36.26 / 0.970	37.33 / 0.980
Rain1200M	26.76 / 0.844	27.07 / 0.846	29.18 / 0.874	31.23 / 0.896
Rain100H	29.12 / 0.898	28.58 / 0.892	28.93 / 0.894	28.99 / 0.895
SPA-data	32.89 / 0.939	33.68 / 0.943	33.86 / 0.947	34.26 / 0.951

TABLE VIII: Analysis of hyper-parameter λ in training on the 1400-100L-1200M-100H sequence using MPRNet [29].

	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 2.0$	$\lambda = 5.0$
Rain1400	29.91 / 0.901	29.80 / 0.902	30.76 /0.911	29.75 / 0.899	29.82 / 0.897
Rain100L	37.08 / 0.975	37.19 / 0.976	37.33 / 0.980	37.10 / 0.976	37.07 / 0.973
Rain1200M	28.35 / 0.848	29.41 / 0.860	31.23 / 0.896	30.26 / 0.876	28.47 / 0.861
Rain100H	28.62 / 0.892	28.79 / 0.895	28.99 / 0.895	28.74 / 0.892	28.73 / 0.891
SPA-data	34.03 / 0.947	34.07 / 0.948	34.26 / 0.951	33.84 / 0.943	33.70 / 0.942

TABLE IX: Stability analysis of the framework. Mean and standard deviation of performance over 10 runs are reported.

Training Sequence	Methods	Avg Memory		SPA-data	
		PSNR	SSIM	PSNR	SSIM
1400-1200M-100H-100L	MFDNet	31.93±0.05	0.923±0.002	34.12±0.02	0.949±0.001
	Restormer	32.48±0.06	0.927±0.002	34.04±0.01	0.948±0.000
	MPRNet	31.69±0.07	0.916±0.001	33.54±0.02	0.946±0.001

TABLE X: Comparison with CL methods using MFDNet [13]

Training Sequence	Metrics	EWC	MAS	LwF	CLGID	CLGID†
1400-1200M-100H-100L	Avg Mem	24.73	23.23	24.05	31.94	31.58
	SPA-data	28.19	27.77	28.44	34.12	34.06
1400-100L-1200M-100H	Avg Mem	23.74	26.68	23.03	31.72	31.26
	SPA-data	28.51	28.41	27.43	34.42	34.24
100L-100H-1400-1200M	Avg Mem	26.33	27.18	20.87	31.27	30.91
	SPA-data	26.87	28.13	26.45	34.20	34.09
100H-100L-1400-1200M	Avg Mem	22.26	23.03	25.32	31.32	31.09
	SPA-data	28.26	28.24	29.56	33.96	33.82

increases, CLGID$^\diamondsuit$ achieves an average reduction of over 50% in GAN training FLOPs, training time, and parameter storage. Meanwhile, CLGID¶ reduces the number of GAN inference calls required for replay by an average of 26.9%, thereby decreasing the overall replay-related FLOPs and time consumption. In terms of performance, CLGID$^\diamondsuit$ exhibits a slight degradation in memory and generalization, but it still significantly outperforms all other SOTA methods. CLGID¶ maintains nearly the same level of memory and generalization as the original CLGID, as the replay data reuse strategy does not incur meaningful information loss during training.

Validation on each framework component. To validate the effectiveness of each loss function, we train our method under circumstances that remove replay loss and consistency loss in a successive manner. The results are shown in Table VII, and it is evident that each loss contributes to the promotion of memory performance on training datasets and generalization performance on the real-world dataset. The proposed CLGID achieves the best performance.

Validation on hyper-parameter λ. We conduct ablation studies to verify the hyper-parameter λ of balancing the two loss terms: the interleave loss, and consistency loss. The experiment results are illustrated in Table VIII. We found that excessively large or small hyper-parameter can lead to an imbalance between the two losses during optimization, resulting in a decrease in memory and generalization performance. Considering the trade-off, we ultimately set λ to 1 through a comprehensive search, thereby achieving optimal memory and generalization performance.

Validation on framework stability. To validate the stability of the proposed framework, we conduct experiments using all three de-raining networks and perform validation under one four-dataset stream (1400-1200M-100H-100L). We perform 10 independent runs and report the mean and standard deviation of PSNR and SSIM for both memory and generalization performance. As shown in Tab. IX, the standard deviation of PSNR remains below 0.10 dB, and that of SSIM is less than 0.003 across all settings. These variations are considerably smaller than the observed performance differences between CLGID and competing methods, indicating that the influence of randomness is negligible. The results confirm that our framework is stable.

Comparison with general CL methods. Table X reports the results of three representative general continual learning (CL) approaches, including EWC [54], MAS [66], and LwF [67], on a stream of four datasets. All three show clear performance drops compared with CLGID. The main reason is that these general CL methods are not customized for the low-level de-raining task, making them poorly matched to its characteristics. EWC and MAS estimate parameter importance via Fisher information or output sensitivity, under assumptions suited to preserving class-discriminative features. In de-raining, however, performance depends on many dense, pixel-wise filters that model rain orientation, density, and scale; these estimates become noisy and either over-constrain adaptation or fail to protect critical filters, causing forgetting. While both LwF and CLGID adopt distillation, the key difference is the choice of inputs. LwF operates on current inputs that scarcely activate knowledge associated with previous degradations, providing limited constraint on their parameters and thus allowing forgetting. CLGID instead distills on replayed inputs embedding previous degradations, ensuring strong activation of the relevant pathways to retain prior capabilities while accommodating new backgrounds.

Discussion on the effect of training order. To examine the potential influence of dataset order, we conduct experiments on 12 permutations of four datasets and evaluate generalization performance on SPA-data, with rain severity approximately ranked as 1400 < 100L < 1200M < 100H. Several tendencies emerge from the results, as shown in Table 10. First, orders progressing from lighter to heavier rain often achieved higher scores, e.g., 1400-100L-1200M-100H (34.42 dB) vs. 100H-1400-1200M-100L (33.89 dB), which may relate to curriculum learning, where gradually increasing task difficulty helps the model build stable feature representations. Second, large fluctuations in dataset difficulty within the sequence may coincide with performance drops. Finally, starting with the difficult dataset also tend to weaken generalization, e.g., 100H-100L-

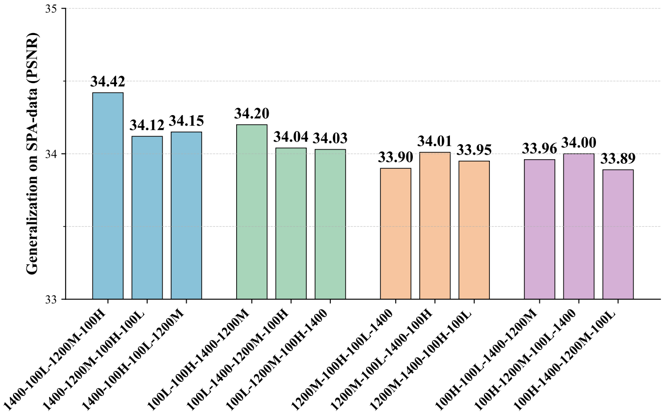


Fig. 10: Generalization performance (PSNR on SPA-data) for 12 training orders of four datasets using MFDNet [13], with bars grouped by the first dataset in the sequence.

TABLE XI: Comparison with the Mix-up strategy using MFDNet [13]. Results of CLGID and CLGID† are averaged over the four training sequences in Table I.

Methods	Rain1400	Rain1200M	Rain100H	Rain100L	SPA-data
Individual	31.79 / 0.920	32.24 / 0.920	27.70 / 0.886	36.16 / 0.978	–
CLGID	31.24 / 0.918	31.82 / 0.910	27.15 / 0.869	36.04 / 0.977	34.18 / 0.950
CLGID†	31.11 / 0.914	31.10 / 0.901	26.90 / 0.857	35.73 / 0.975	34.05 / 0.948
Mix-up	32.03 / 0.925	31.10 / 0.900	25.71 / 0.840	34.53 / 0.961	33.76 / 0.947

1400-1200M (33.96 dB). While exceptions exist, these trends suggest that smooth, curriculum-like order can be optimal.

Comparison with mix-up strategy. We further evaluate a mixed training strategy (Mix-up), where the de-raining network is trained on a combined set of multiple datasets rather than sequentially. As shown in Table XI, Mix-up yields higher performance on Rain1400 but drops on the other datasets and shows inferior generalization to SPA-data. This is attributed to severe data imbalance: Rain1400 contains far more samples, causing its gradients to dominate optimization and bias the solution toward its domain. Consequently, the final model under-represents smaller datasets in the learned feature space, memorizes the dominant domain while failing to integrate knowledge from under-represented ones, and this insufficient integration across datasets results in degraded generalization to real-world rainy scenes.

V. Conclusion

This paper presents a new generalized de-raining framework, CLGID, which empowers de-raining networks to accumulate knowledge from increasingly abundant de-raining datasets, improving their ability to generalize to unseen real-world scenes. Our inspiration stems from the human brain's complementary learning system, which enables humans to constantly learn and memorize a stream of perceived events and gradually acquire the generalization ability to unseen situations across memorized events. This remarkable human ability closely aligns with our research goals. Extensive experiments validate that CLGID effectively accumulate de-raining knowledge and delivers superior de-raining generalization performance in unseen real-world rainy scenes.

References

[1] K. Wang, X. Fu, Y. Huang, C. Cao, G. Shi, and Z.-J. Zha, "Generalized uav object detection via frequency domain disentanglement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[2] K. Wang, X. Fu, C. Ge, C. Cao, and Z.-J. Zha, "Towards generalized uav object detection: A novel perspective from frequency domain disentanglement," *International Journal of Computer Vision*, pp. 1–29, 2024.

[3] R. Xu, C. Wang, J. Zhang, S. Xu, W. Meng, and X. Zhang, "Rssformer: Foreground saliency enhancement for remote sensing land-cover segmentation," *IEEE Transactions on Image Processing*, 2023.

[4] R. Xu, Y. Li, C. Wang, S. Xu, W. Meng, and X. Zhang, "Instance segmentation of biological images using graph convolutional network," *Engineering Applications of Artificial Intelligence*, 2022.

[5] R. Xu, C. Wang, S. Xu, W. Meng, and X. Zhang, "Dc-net: Dual context network for 2d medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention*, 2021.

[6] J. Zhang, K. Wang, S. Wang, M. Li, H. Liu, S. Wei, Z. Wang, Z. Zhang, and H. Wang, "Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks," *arXiv preprint arXiv:2412.06224*, 2024.

[7] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "Navid: Video-based vlm plans the next step for vision-and-language navigation," *arXiv preprint arXiv:2402.15852*, 2024.

[8] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[9] Y. Wang, Y. Song, C. Ma, and B. Zeng, "Rethinking image deraining via rain streaks and vapors," in *European Conference on Computer Vision*, 2020.

[10] R. Li, L.-F. Cheong, and R. 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*, 2019.

[11] K. Jiang, Z. Wang, P. Yi, C. Chen, G. Wang, Z. Han, J. Jiang, and Z. Xiong, "Multi-scale hybrid fusion network for single image deraining," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 7, pp. 3594–3608, 2021.

[12] Z. Huang and J. Zhang, "Contrastive unfolding deraining network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5155–5169, 2022.

[13] 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.

[14] W. Yang, R. T. Tan *et al.*, "Deep joint rain detection

and removal from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[15] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, 2019.

[16] M. Zhou, J. Xiao, Y. Chang, X. Fu, A. Liu, J. Pan, and Z.-J. Zha, "Image de-raining via continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[17] J. Xiao, M. Zhou, X. Fu, A. Liu, and Z.-J. Zha, "Improving de-raining generalization via neural reorganization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[18] M. Liu, W. Yang, Y. Hu, and J. Liu, "Dual prompt learning for continual rain removal from single images," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023.

[19] X. Fu, J. Huang *et al.*, "Removing rain from single images via a deep detail network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[20] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[21] 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.

[22] X. Fu, J. Xiao, Y. Zhu, A. Liu, F. Wu, and Z.-J. Zha, "Continual image deraining with hypergraph convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 9534–9551, 2023.

[23] D. Cheng, Y. Ji, D. Gong, Y. Li, N. Wang, J. Han, and D. Zhang, "Continual all-in-one adverse weather removal with knowledge replay on a unified network structure," *IEEE Transactions on Multimedia*, vol. 26, pp. 8184–8196, 2024.

[24] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory." *Psychological Review*, 1995.

[25] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? complementary learning systems theory updated," *Trends in Cognitive Sciences*, 2016.

[26] B. L. McNaughton and R. G. Morris, "Hippocampal synaptic enhancement and information storage within a distributed memory system," *Trends in Neurosciences*, 1987.

[27] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, 1999.

[28] 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.

[29] 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.

[30] J. L. McClelland, "Incorporating rapid neocortical learning of new schema-consistent information into complementary learning systems theory." *Journal of Experimental Psychology: General*, 2013.

[31] D. Tse, T. Takeuchi, M. Kakeyama, Y. Kajii, H. Okuno, C. Tohyama, H. Bito, and R. G. Morris, "Schema-dependent gene activation and memory encoding in neocortex," *Science*, 2011.

[32] J.-H. Kim, C. Lee, J.-Y. Sim, and C.-S. Kim, "Single-image deraining using an adaptive nonlocal means filter," in *2013 IEEE International Conference on Image Processing*, 2013.

[33] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[34] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, 2011.

[35] R. Yasarla, V. A. Sindagi, and V. M. Patel, "Syn2real transfer learning for image deraining using gaussian processes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[36] Y. Liu, Z. Yue, J. Pan, and Z. Su, "Unpaired learning for deep image deraining with rain direction regularizer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[37] Y. Ye, C. Yu, Y. Chang, L. Zhu, X.-l. Zhao, L. Yan, and Y. Tian, "Unsupervised deraining: Where contrastive learning meets self-similarity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[38] Y. Zheng, X. Yu, M. Liu, and S. Zhang, "Single-image deraining via recurrent residual multiscale networks," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 3, pp. 1310–1323, 2020.

[39] Y. Chang, M. Chen, C. Yu, Y. Li, L. Chen, and L. Yan, "Direction and residual awareness curriculum learning network for rain streaks removal," *IEEE transactions on neural networks and learning systems*, 2023.

[40] Y.-T. Wang, X.-L. Zhao, T.-X. Jiang, L.-J. Deng, Y. Chang, and T.-Z. Huang, "Rain streaks removal for single image via kernel-guided convolutional neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3664–3676, 2020.

[41] B. Li, X. Liu, P. Hu, Z. Wu, J. Lv, and X. Peng, "All-in-one image restoration for unknown corruption," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 17 452–17 462.

[42] Y. Gou, B. Li, Z. Liu, S. Yang, and X. Peng, "Clearer: Multi-scale neural architecture search for image restora-

tion," *Advances in neural information processing systems*, vol. 33, pp. 17 129–17 140, 2020.

[43] B. Li, H. Zhao, W. Wang, P. Hu, Y. Gou, and X. Peng, "Mair: A locality-and continuity-preserving mamba for image restoration," *arXiv preprint arXiv:2412.20066*, 2024.

[44] Y. He, A. Jiang, L. Jiang, L. Peng, Z. Wang, and L. Wang, "Dual-path coupled image deraining network via spatial-frequency interaction," in *2024 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2024, pp. 1452–1458.

[45] W. Li, G. Chen, and Y. Chang, "An efficient single image de-raining model with decoupled deep networks," *IEEE Transactions on Image Processing*, vol. 33, pp. 69–81, 2023.

[46] A. Sivaanpu and K. Thanikasalam, "A dual cnn architecture for single image raindrop and rain streak removal," in *2022 7th International Conference on Information Technology Research (ICITR)*. IEEE, 2022, pp. 1–6.

[47] J. Liu, Q. Li, X. Min, Y. Su, G. Zhai, and X. Yang, "Pixel-learnable 3dlut with saturation-aware compensation for image enhancement," *IEEE Transactions on Multimedia*, 2024.

[48] S. Li, W. Ren, F. Wang, I. B. Araujo, E. K. Tokuda, R. H. Junior, R. M. Cesar-Jr, Z. Wang, and X. Cao, "A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives," *International Journal of Computer Vision*, 2021.

[49] W. Yang, R. T. Tan, S. Wang, Y. Fang, and J. Liu, "Single image deraining: From model-based to data-driven and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[50] Y. Gu, C. Wang, and J. Li, "Incremental image de-raining via associative memory," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[51] X. Kong, C. Dong, and L. Zhang, "Towards effective multiple-in-one image restoration: A sequential and prompt learning strategy," *arXiv preprint arXiv:2401.03379*, 2024.

[52] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[53] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[54] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[55] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.

[56] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 4, pp. 4396–4415, 2022.

[57] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, "Deep domain generalization via conditional invariant adversarial networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 624–639.

[58] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2100–2110.

[59] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2229–2238.

[60] P. Chattopadhyay, Y. Balaji, and J. Hoffman, "Learning to balance specificity and invariance for in and out of domain generalization," in *European Conference on Computer Vision*. Springer, 2020, pp. 301–318.

[61] H. Wang, Z. Yue, Q. Xie, Q. Zhao, Y. Zheng, and D. Meng, "From rain generation to rain removal," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[62] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2005.

[63] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, 1951.

[64] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics Letters*, 2008.

[65] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, 2004.

[66] R. Aljundi, K. Kelchtermans, and T. Tuytelaars, "Task-free continual learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 254–11 263.

[67] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.