Dual Prompt Learning for Continual Rain Removal from Single Images

Minghao Liu\textsuperscript{1,2,3}, Wenhan Yang\textsuperscript{4}, Yuzhang Hu\textsuperscript{1} and Jiaying Liu\textsuperscript{1}\textsuperscript{*}
\textsuperscript{1}Wangxuan Institute of Computer Technology, Peking University
\textsuperscript{2}School of EECS, Peking University
\textsuperscript{3}School of CS, Peking University
\textsuperscript{4}Peng Cheng Laboratory
lmlhmh@stu.pku.edu.cn, yangwh@pcl.ac.cn, \{yuzhanghu, liuiaying\}@pku.edu.cn

Abstract

Recent efforts have achieved remarkable progress on single image deraining on the stationary distributed data. However, catastrophic forgetting raises practical concerns when applying these methods to real applications, where the data distributions change constantly. In this paper, we investigate the continual learning issue for rain removal and develop a novel efficient continual learned deraining transformer. Different from the typical replay or regularization-based methods that increase overall training time or parameter space, our method relies on compact prompts which are small learnable parameters, to maintain both task-invariant and task-specific knowledge. Our prompts are applied at both image and feature levels to leverage effectively transferred knowledge of images and features among different tasks. We conduct comprehensive experiments under widely-used rain removal datasets, where our proposed dual prompt learning consistently outperforms prior state-of-the-art methods. Moreover, we observe that, even though our method is designed for continual learning, it still achieves superior results on the stationary distributed data, which further demonstrates the effectiveness of our method. Our website is available at: http://liuminghao.com.cn/DPL/.

1 Introduction

As one of the most common weather degradation, rain streaks heavily reduce visibility and corrupt the information captured by images, impacting both human visual experience and computer vision algorithms like detection [Carion et al., 2020], segmentation [Chen et al., 2018], and depth estimation [Wang et al., 2020], which is closely related to many practical applications, e.g., autonomous navigation and surveillance systems.

In recent years, remarkable progress has been achieved in single image deraining, especially for deep learning based methods. Many methods built on convolutional networks [Fu et al., 2017; Ren et al., 2019; Zamir et al., 2021] continue to break new ground in performance. Recently, transformer-based methods [Valanarasu et al., 2022] have achieved excellent results. However, rain degradation is complex and diverse, and most existing models only learn fixed mappings between paired rainy and clean images. As a result, deep neural networks may lose previously acquired knowledge and experience performance decline when faced with changing data distributions, limiting their real-world applicability.

To alleviate this issue, continual learning methods are developed to overcome catastrophic forgetting. They can be mainly divided into three categories: parameter isolation-based methods [Zhang et al., 2020; Mallya et al., 2018; Xu and Zhu, 2018], replay-based mechanisms [Zhao et al., 2021; Kirkpatrick et al., 2017; Pascanu and Bengio, 2013; Aljundi et al., 2018], and regularization-based methods [Li and Hoiem, 2017; Zenke et al., 2017]. However, parameter isolation and replay-based methods are computationally expensive, while regularization-based methods become inefficient and the parameters increase linearly as tasks increase.

--

\textsuperscript{*}Corresponding Author.
In addition, all three kinds of methods fail to automatically select relevant knowledge components for arbitrary samples without knowing their task identity, which is a tremendous limitation in real applications.

Lately, researchers have drawn inspiration from recent advances in prompt-based learning (prompting) [Liu et al., 2021], which reformulates learning tasks by designing prompts [Raffel et al., 2020] instead of directly adapting model weights. Prompts encode the knowledge specifically related to a particular task and allow for more efficient use of a pre-trained frozen model than fine-tuning, which provides an ideal mechanism [Wang et al., 2022] for continual learning.

In this paper, we propose a novel continual learning scheme for single image deraining called Dual Prompt Learning (DPL), which is orthogonal to existing replay-based and regularization-based methods. Fig. 1 demonstrates its effectiveness. It does not need to know the task identity or boundaries, therefore more applicable to real scenarios. Its objective is to learn to select and update compact prompts, i.e., small learnable parameters, to instruct the transformer-based rain removal model to maintain both task-invariant and task-specific knowledge. In detail, a subset of prompts are selected from the image/feature prompt pools based on our proposed instance-wise query mechanism and are concatenated with input image/embedded tokens for further processing. These prompts applied at both image and feature levels effectively leverage transferred knowledge of these two levels jointly. Furthermore, a regularization technique is further adopted to penalize the intense changes of important parameters jointly with the learned prompts.

In summary, our work has the following contributions:

• We propose a novel prompt learning-based continual learning (CL) scheme to handle different types of rain streaks with a single model. To the best of our knowledge, it is the first time to apply this new kind of CL method for low-level vision, which leads to superior performance on various benchmarks.

• We develop a dual prompt learning method for deraining, where prompts are applied at both image and feature levels to leverage effectively transferred knowledge of images and features jointly.

• Our DPL is further augmented by parameter regularization. The joint regularization of model parameters and learnable prompts obviously further improves the performance.

• Even though our method is designed for the continual learning scenario, it achieves competitive results against state-of-the-art methods on the stationary distributed data.

2 Related Work

2.1 Single Image Rain Removal

Recently, there have been several methods that have achieved significant progress in single image rain removal. Yang et al. [Yang et al., 2017], and Fu et al. [Fu et al., 2017] proposed the first framework on the rain streak removal. In [Yang et al., 2017], the rain streak detection and removal are modeled in the multi-task manner. In [Fu et al., 2017], the residual learning is applied for rain removal. Later works are proposed with more complex architectures to improve rain removal performance, including joint rain density estimation and deraining [Ren et al., 2019], non-local operation-based encoder-decoder network [Li et al., 2018a], multi-stage network [Zhang and Patel, 2018], conditional generative adversarial network [Zhang et al., 2019], and deep convolutional and recurrent neural network [Li et al., 2018b] that removes rain streaks stage by stage, etc. Recently, transformer-based approaches [Valanarasu et al., 2022] have been explored for weather removal tasks, demonstrating their superior performance compared to convolutional networks. However, all the mentioned deep learning-based methods suffer from catastrophic forgetting issue, failing to maintain their effectiveness when applied to different types of rainy datasets/tasks in real applications. Comparatively, our work explores the continual learning approach for image rain removal to adapt the model to a series of rain streaks.

2.2 Continual Learning

A large amount of research in continual learning follows a learning paradigm that involves continuously adapting the model weights, either partially or fully, as the data distribution changes [De Lange et al., 2021; Mai et al., 2022]. These approaches focus on preserving previous knowledge while also adapting to shifting data. The methods of overcoming catastrophic forgetting can be mainly divided into three categories: replay-based mechanisms [Zhao et al., 2021; Kirkpatrick et al., 2017; Pascancu and Bengio, 2013; Aljundi et al., 2018], regularization-based methods [Li and Hoiem, 2017; Zenke et al., 2017], and parameter isolation-based methods [Zhang et al., 2020; Mallya et al., 2018; Xu and Zhu, 2018]. In detail, replay-based and parameter isolation-based methods are computationally expensive since they require recording the old tasks’ targets and computing old tasks’ forward pass process for each novel data sample. Regularization-based methods are cost-effective. The representative form of the classic regularization-based method is Elastic Weight Consolidation (EWC) [Kirkpatrick et al., 2017], which quantifies how essential each parameter is for a task with the diagonal of the Fisher information matrix[Pascancu and Bengio, 2013] and protects critical weights with an additional regularization to restrict their movement when updating for the new job. Further, memory-aware synapses (MAS) [Aljundi et al., 2018] compute the parameter importance based on how sensitive the predicted output function is to a change in this parameter, and penalize changed essential parameters. Zhou et al. [Zhou et al., 2021] proposed a first and second-order parameter importance to jointly estimate the status of one parameter. In this paper, we propose a novel continual learning scheme for single image deraining, which is orthogonal to existing replay-based and regularization-based methods.

2.3 Prompting for Transfer Learning

Prompting is applying a function to modify the input so that a model gets additional information about the task.
However, the design of a prompting function is challenging and requires heuristics. Recent works in NLP, including prompt tuning [Lester et al., 2021] and prefix tuning [Li and Liang, 2021], seek to address this issue by applying learnable prompts in a continuous space, achieving excellent performance for transfer learning. Prompts capture dataset-specific knowledge with much smaller additional parameters than its competitors, such as Adapter [Peiffer et al., 2020] and LoRA [Hu et al., 2021], which achieve remarkable performance in many downstream tasks. Prompt techniques have acted out fantastic value for transfer learning. Recently, Wang et al. [Wang et al., 2022] revealed its significance to continual learning problems. In our work, prompt learning is introduced to construct a continual rain removal method.

3 Dual Prompt Learning for Continual Rain Removal

Prompt learning is a recent popular technique for transfer learning and model adaptation. It uses task-specific prompts functions, which improves sequential modeling capacity for continuous feature learning. In this paper, we propose to utilize two-level prompt pools that lead to superior performance even without knowing task identities, as shown in Fig. 2. Instead of using the naive way of prompt learning that relies on task identities and lacks the flexibility to distinguish task-independent/relevant knowledge, the prompt pools enable an online search for the prompts to share the knowledge when tasks are similar and maintain independent knowledge otherwise. This prompt learning based on prompt pools is applied at both image and feature levels, to jointly leverage transferred knowledge of two stages.

3.1 Definition and Formulation

We define a sequence of $T$ de-raining tasks as $\mathcal{D} = \{D_1, \ldots, D_T\}$ where the $t$-th task $D_t = \{(x^t_i, y^t_i)\}_{i=1}^{n_t}$ contains $n_t$ pairs of samples, where each pair consists of a rainy image $x^t_i \in \mathcal{X}$ and its clean corresponding background $y^t_i \in \mathcal{Y}$. Our goal is to train a model $f(\theta_{\text{model}}) : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by $\theta_{\text{model}}$, where $\mathcal{X}$ is the rain images domain and $\mathcal{Y}$ is the corresponding clean images domain. However, the data of $D_1, D_2, \ldots, D_n$ is not available when training $D_{n+1}$.

Given the input image $x \in \mathbb{R}^{C \times H \times W}$ where $C$ is the channel number, and $H \times W$ is the size of the image. We build a transformer-based deraining backbone $f = f_r \circ f_e$, where $f_r$ is the input embedding layer, and $f_e$ represents a stack of self-attention layers and the subsequent rain removal network.

3.2 Image-Level Prompt-Based Learning

To alleviate the deficiency of linear growth of space expenditure caused by the replay mechanism, and transfer the image-level prior knowledge among different tasks effectively, we integrate image-level prompt learning into our proposed continual learning model. An image prompt pool is defined as $\mathcal{P} = \{P_1, \ldots, P_M\}$ containing a certain amount of trainable prompts $P_t = (K_t, V_t)$ where value $V_t \in \mathbb{R}^{C \times H \times W}$ is a representation with the same size as the input image, and key $K_t = f_e(V_t)$ is the embedded value to match the key of input. Following the notations in Sec. 3.1, $x$ and $x_e = f_e(x)$ are the input and its corresponding embedding feature, respectively. Note that we omit the task index $t$ in our notation as our method is general enough to be applied even without the task identity. Ideally, the input $x$ instance itself decides which prompts to be chosen through matching. To this end, we utilize a distance-measuring function $\gamma$ to measure the similarity of a prompt and the input image. We directly take $\gamma$ as cosine distance, which is proven as a good choice empirically [Wang et al., 2022]. During the training of task $n$, we maintain a prompt frequency table $Q_n = [q^n_1, q^n_2, \ldots, q^n_M]$, where each entry represents the normalized frequency of prompt $P_t$ being selected in task $n$. As the feature includes richer semantic information, the similarity is calculated at the feature level to better reflect more intrinsic information instead of the pixel level. During the training of task $n$, given an input $x$, we lookup the top-$N$ prompts by simply solving the following objective:

$$\mathcal{P}_x = \arg \min \sum_{i=1}^{N} \gamma(K_{s_i}, f_e(x)) \cdot q^n_{s_i},$$

where $\mathcal{P}_x$ represents the subset of top-$N$ prompts selected specifically for $x$ from $\mathcal{P}$, and the prompt frequency table actually encourages the choice of diversified prompts, which will be removed during the testing process.

After selection, we concatenate the top-$N$ prompts with the input along the channel dimension and put them into the embedding layer together. These prompts have the same size as the input and thus they can jointly encode knowledge with the input in model training.

$$x_1 = [V_{s_1}; V_{s_2}; \cdots; V_{s_N}; x],$$

where $[;]$ represents concatenation along the channel dimension.

3.3 Feature-Level Prompt-Based Learning

Given the input $x$, and the transformer-based model $f = f_r \circ f_e$. The embedding layer $f_e : \mathbb{R}^{C \times H \times W} \rightarrow \mathbb{R}^{L \times D}$ projects the patched image to the embedding feature $x_e = f_e(x) \in \mathbb{R}^{L \times D}$ where $L$ is the length of a token and $D$ is the embedding dimension. At the feature level, we also maintain a set of prompts $\mathcal{P}' = \{P'_1, P'_2, \ldots, P'_M\}$ containing a certain amount of prompts $P'_t = (K'_t, V'_t)$ where $K'_t = V'_t \in \mathbb{R}^{L \times D}$, for $i = 1, 2, \ldots, M$, and its corresponding prompt frequency table for task $n$, $Q'_n = [q'^n_1, q'^n_2, \ldots, q'^n_M]$, where each entry represents the normalized frequency of prompt $P'_t$ being selected in task $n$. Similar to Sec. 3.2, we denote $\gamma$ as a cosine distance function to score the match between $x_e = f_e(x)$ and prompts $P'_{s'_1}, P'_{s'_2}, \ldots, P'_{s'_N}$. In other words, during task $n$, we lookup the top-$N$ prompts by simply solving the following objective:

$$\mathcal{P}'_x = \arg \min \sum_{i=1}^{N} \gamma(K'_{s'_i}, f_e(x)) \cdot q'^n_{s'_i},$$

where $\{s'_1, s'_2, \ldots, s'_N\}$ is a subset of $N$ indices from $[1, M]$, and we can then adapt the input embedding as follows:

$$x_2 = [V'_{s'_1}; V'_{s'_2}; \cdots; V'_{s'_N}; f_e(x_1)], 1 \leq N \leq M.$$
where $[\cdot]$ represents concatenation along the token length dimension. In Eqn. (4), $x_1$ represents the synthesized input in Sec. 3.2, and note that when selecting these top-$N$ prompts, we use $x$, but during training, we concatenate these prompts with $x_1 = [V_{s_1}^x; V_{s_2}^x; \cdots; V_{s_N}^x; x]$. After concatenation, $x_2$ is put into the subsequent rain-removal network for further training. As in Sec. 3.2, these prompts are also trainable.

### 3.4 Joint Regularization of Parameters and Prompts

When the neural network is trained on Task $n$ and Task $n+1$ sequentially, we hope the network can still maintain the performance of Task $n$. Thus, the regularization is imposed on the model parameters and prompts jointly. On Task $n$, the rain removal model’s parameters are denoted as $\theta_{model}^n = \{\theta_1^n, \theta_2^n, \cdots, \theta_r^n\}$ where $r$ is the depth of the network, and the parameter set of the overall prompts is denoted as $\theta_{prompt}^n = \{\theta_1^{s_1}, \theta_2^{s_2}, \cdots, \theta_s^{s_I}\}$, where $s$ is the total quantity of all parameters pertained to these prompts. We signify $\theta^n = \theta_{model}^n \cup \theta_{prompt}^n$. For the sake of convenience, we imply $\theta^n = \{\theta_1^n, \theta_2^n, \cdots, \theta_r^n\}$. $X^n$ and $V^n$ indicate the actual calculation, and the clean images set on Task $n$, respectively. Suppose $(x, y)$ is a rainy/clean image pair. When it is fed into the network, the degradation of performance on Task $n$ introduced by the training of network on Task $n+1$ can be evaluated as:

$$\Delta f(\theta^{n+1}, \theta^n, x, y) = Dist(f(x, \theta^{n+1}), f(x, \theta^n))$$

$$= \frac{1}{2} \left(\nabla_{\theta^n} l f(x, \theta^{n+1})^T \cdot \delta \theta^n \right) + O(\delta \theta^n)^3$$

A small $I(\theta^n)$ leads to more preservation of the knowledge of Task $n$ and can be adopted as an effective regularization when training on Task $n+1$.
3.5 Optimization Objective Function
At every training step, after selecting $N$ image prompts following the aforementioned objective. The joint input $x_1$ is fed into the embedding layer. After that, $N$ feature prompts are selected and concatenated. The adapted embedding feature $x_2$ is fed into the subsequent layers. Additionally, the prompt distance penalty and the penalty of parameters and prompts are also put into the optimization objective. Overall, on Task $n$, we seek to minimize the end-to-end training loss function:

$$
L_x = \min_{P, P', \theta} \alpha L_1(f(x), y) + \beta L_p(f(x), y) \\
+ \zeta \sum_{P_x} \gamma \left( f_x(P_x) ; f_e(x) \right) \cdot q_{x_i} \\
+ \eta \sum_{P_x} \gamma \left( P_x' ; f_e(x) \right) \cdot q_{x_i}^{n'} + \omega I(\theta^n).
$$

(9)

where $L_1$ and $L_p$ represents smooth $L_1$ loss and perceptual loss, respectively. The third and fourth items are surrogate losses to pull selected prompts closer to the corresponding query features and the fifth item is the parameter penalty aforementioned in Eqn. (8). $\alpha, \beta, \zeta, \eta, \omega$ are parameters that control the importance of each term.

3.6 Learning of Prompt Pools
For each task, $P$ and $P'$ are initialized randomly. In detail, for a $P_i = (K_i, V_i), 1 \leq i \leq M$, $V_i$ is randomly initialized and has the same size as the input image, and $K_i$ is calculated by $K_i = f_e(V_i)$, where $f_e$ is the embedding layer of a particular model. Similarly, for a $P_i' = (K_i', V_i')$, $1 \leq i \leq M$, $V_i'$ is also randomly initialized and has the same size as the token, and $K_i' = V_i'$ is the same as $V_i$. During each training epoch, after we draw a mini-batch $B = \{(x_i^t, y_i^t)\}_{i=1}^l$, where $l$ is the batch size, two sets of chosen prompts $P_B, P_B'$ are respectively maintained for the dual prompt pools. Initially, they are both empty sets. For each pair of rainy/clean images, $(x, y)$ in $B$, top-$N$ image prompts $P_x$ and top-$N$ feature prompts $P_x'$ are obtained by solving Eqn. (1) and Eqn. (3). We update the set of chosen prompts by $P_B = P_B \cup P_x, P_B' = P_B' \cup P_x'$, and update $Q_t$ and $Q_t'$ by adding 1 to the frequency-items corresponding to the selected prompts. Then, we calculate the sample loss $L_2$ of every input in the mini-batch by Eqn. (9). We calculate per batch loss $L_B$ by accumulating $L_2$, and we update the parameters in $P$ and $P'$ by back-propagation immediately. In other words, $P$ and $P'$ are updated once for each epoch. The whole process of our proposed DPL for continual rain removal is summarized in Algorithm 1.

4 Experiment Results
In order to fully evaluate the capabilities of our proposed continual learning scheme, we integrate it with a state-of-the-art rain removal baseline TransWeather [Valanarasu et al., 2022]. Through extensive experimentation on several widely-used rain removal datasets, our DPL method consistently demonstrates superior performance in comparison to other existing continual learning methods. In particular, the results of all experiments indicate that our method performs excellently in both the capacities to adapt to new tasks and maintain high performance on previous ones.

### Algorithm 1 Dual Prompt Learning
**Require:** DataSet $D = \{D_1, D_2, \cdots, D_T\}$, where $D_t$ includes an amount of pairs of rainy/clean images $(\{x_i, y_i\})_{i=1}^L$ and $T$ is the number of tasks.
**Parameter:** Network $f$ and overall prompts $P, P'$ parameterized by $\theta$, image-level prompt pool $P = \{P_1, \cdots, P_M\}$, feature-level prompt pool $P' = \{P_1', \cdots, P_M'\}$, where $P_i = (K_i, V_i), P_i' = (K_i', V_i')$, for $1 \leq i \leq M$. Maintained frequency set $Q_t = [q_1, q_2, \cdots, q_M], Q_t' = [q_1', q_2', \cdots, q_M']$ for the $t$-th task, number of training epochs of the $t$-th task $E_t, 1 \leq t \leq T$, learning rate $\rho$, hyperparameters of loss function $\alpha, \beta, \zeta, \eta, \omega$.

**Begin**
1: Load the pre-trained VGG-16 model, and get $D = \{D_1, D_2, \cdots, D_T\}$.
2: for $t = 1, 2, \cdots, T$ do
   3: Initialize $Q_t$ and $Q_t'$ with real number $1$. Initialize $P$ and $P'$ with random prompt-sized items.
   4: for $(x, y)$ in $D_B$ do
      5: Draw a mini-batch $B = \{(x_i^t, y_i^t)\}_{i=1}^l$. Initialize the sets of chosen prompts for this batch: $P_B = \{\}$, $P_B' = \{\}$.
      6: for $(x, y)$ in $B$ do
          7: Calculate the embedding input $x_e = f_e(x)$.
          8: Lookup top-$N$ image prompts by solving Eqn. (1).
          9: Prepending $x_e$ with corresponding top-$N$ prompts by $x_1 = [V_1; V_2; \cdots; V_N; x_e]$.
          10: Calculate the embedding input $f_e(x_1)$.
          11: Lookup top-$N$ feature prompts by solving Eqn. (3).
          12: Prepending $x_1$ with corresponding top-$N$ prompts by $x_2 = [V_1'; V_2'; \cdots; V_N' ; f_e(x_1)]$.
          13: Calculate the rain-free result $f(x) = f_r(x_2)$.
          14: Calculate per sample loss by solving Eqn. (9).
          15: Update sets of chosen prompts: $P_B = P_B \cup P_x, P_B' = P_B' \cup P_x'$.
          16: Update sets of chosen prompts: $P_B = P_B \cup P_x', P_B' = P_B' \cup P_x'$.
          17: Update $Q_t$ and $Q_t'$, by adding 1 to the frequency-items corresponding to the selected prompts.
          18: end for
      19: Calculate per batch loss $L_B$ by accumulating $L_2$.
      20: Update $P$ and $P'$.
      21: end for
      22: Update $\theta$.
      23: end for 0
**End**

4.1 Dataset and Performance Metrics
We evaluate our proposed continual learning scheme on three widely-used rain removal datasets, including Rain100H [Yang et al., 2019], Rain100L [Yang et al., 2019], and Rain800 [Zhang et al., 2019]. In detail, the model is trained on Rain800 (Task 1) and Rain100H (Task 2) sequentially, denoted as Rain800-Rain100H. In addition to the continual task sequence Rain800-Rain100H, we further experiment with continual task sequences Rain800-Rain100L. Both Rain100H and Rain100L consist of 1,800 rainy/clean image pairs for training and 100 pairs for testing while Rain800 possesses 600 training samples and 200 testing images. Peak-Signal-to-Noise Ratio (PSNR) and Structure SIMilarity (SSIM) [Wang et al., 2004] are employed for evaluating the model performance.
Figure 3: Visual comparison of rain streak removal results generated from the continual learning process using baseline. (a) Input: rainy images from Rain800; (b) Task 0: train and test on Rain800; (c) Task 1 with SI: train on Rain800-Rain100H sequentially and independently (SI) and test on Rain800; (d) Task 1 with replay: train on Rain800-Rain100H sequentially with rehearsal and test on Rain800; (e) Task 1 with PIGWM: train on Rain800-Rain100H sequentially with parameter regularization and test on Rain800; (f) Task 1 with DPL: train on Rain800-Rain100H sequentially with dual prompt learning and test on Rain800; (g) GT: clean image.

Table 1: The comparison of performance and additional increased parameter complexity of the continual learning methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Training on Rain800-Rain100H</th>
<th>Performance (PSNR)</th>
<th>Increased Parameter Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (only Rain800)</td>
<td>PSNR 26.63, SSIM 0.8583, Degradation on Rain800 0, 0</td>
<td>22.76</td>
<td>100% + 4% × tasks</td>
</tr>
<tr>
<td>Ours (only Rain800)</td>
<td>PSNR 27.52, SSIM 0.8667, Degradation on Rain800 -0.89, -0.0084</td>
<td>22.48</td>
<td>100%</td>
</tr>
<tr>
<td>SI</td>
<td>PSNR 19.87, SSIM 0.6451, Degradation on Rain800 6.76, 0.2132</td>
<td>22.76</td>
<td>100% + 4% × tasks</td>
</tr>
<tr>
<td>EWC</td>
<td>PSNR 21.64, SSIM 0.7962, Degradation on Rain800 4.99, 0.0621</td>
<td>22.48</td>
<td>100%</td>
</tr>
<tr>
<td>Replay</td>
<td>PSNR 22.51, SSIM 0.8162, Degradation on Rain800 4.12, 0.0421</td>
<td>22.48</td>
<td>100%</td>
</tr>
<tr>
<td>Deep generative</td>
<td>PSNR 22.48, SSIM 0.8058, Degradation on Rain800 4.15, 0.0525</td>
<td>22.48</td>
<td>100%</td>
</tr>
<tr>
<td>PIGWM</td>
<td>PSNR 22.48, SSIM 0.8058, Degradation on Rain800 4.15, 0.0525</td>
<td>22.48</td>
<td>100%</td>
</tr>
<tr>
<td>Ours</td>
<td>PSNR 24.39, SSIM 0.8365, Degradation on Rain800 2.24, 0.0218</td>
<td>22.48</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 2: Comparison of quantitative results in terms of PSNR and SSIM. The models are trained sequentially on task sequence Rain800-Rain100H using continual learning methods. The baseline is trained on Rain800 solely. All the experiments are tested on Rain800.

4.2 Training Details

For a fair comparison, all the parameters setting and training techniques of the baseline model keep consistent with experiments in the original papers. Furthermore, we design distance query function $\gamma$ as cosine distance. For both prompt pools, we assign $M$ to 100, and $N$ to 18, since the mini-batch size is 18. In terms of the hyperparameters, we assign $\alpha$ to 1, $\beta$ to 0.04, $\omega$ to 0.95, and $\zeta$, $\eta$ are both assigned to $1e - 5$. For each task, we train for 50 epochs.

4.3 Results on Benchmark Datasets

To demonstrate the effectiveness of our proposed continual learning algorithm, we conduct both qualitative and quantitative experiments on the above datasets and performance measures. Table 2 and Table 3 report the comprehensive comparison among the baselines, classic continual learning methods, and our method, which indicates that our method can better mitigate catastrophic forgetting on multiple datasets. Surprisingly, Table 2 shows that DPL also achieves competitive results against the baseline method even on a single dataset without applying it to continual learning (PSNR/SSIM: 27.52 dB/0.8667), which further demonstrates the rationality of our method design. Table 4 clearly demonstrates the effectiveness of our method in not only maintaining consistently excellent performance in the previous task but also in achieving satisfactory results in a new task. That is, our method effectively balances the capacity to adapt to new tasks while simultaneously preserving its performance on established ones. As shown in Table 1, dual prompts offer lower parameter complexity compared to traditional continual learning methods, adding only 5.2% extra parameters. By applying parameter
regularization to prompts, our method provides flexible solutions for different scenarios: adding dual prompt pool yields comparable performance to parameter regularization at lower complexity; incorporating parameter regularization can further enhance performance if higher complexity is acceptable.

Table 3: Comparison of quantitative results in terms of PSNR and SSIM. The models are trained sequentially on task sequence Rain800-Rain100L using continual learning methods and tested on Rain800.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Training on Rain800-Rain100L</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
<td>Degradation on Rain800</td>
</tr>
<tr>
<td>Baseline</td>
<td>26.63</td>
<td>0.8583</td>
<td>0, 0</td>
</tr>
<tr>
<td>SI</td>
<td>20.42</td>
<td>0.5823</td>
<td>6.21, 0.2760</td>
</tr>
<tr>
<td>EWC</td>
<td>23.11</td>
<td>0.7840</td>
<td>3.52, 0.0743</td>
</tr>
<tr>
<td>Replay</td>
<td>23.20</td>
<td>0.7758</td>
<td>3.43, 0.0825</td>
</tr>
<tr>
<td>Deep generative</td>
<td>22.25</td>
<td>0.7462</td>
<td>4.38, 0.1121</td>
</tr>
<tr>
<td>PIGWM</td>
<td>23.98</td>
<td>0.8049</td>
<td>2.65, 0.0534</td>
</tr>
<tr>
<td>Ours</td>
<td>24.79</td>
<td>0.8382</td>
<td>1.84, 0.0201</td>
</tr>
</tbody>
</table>

Table 4: Comparison of quantitative results in terms of PSNR and SSIM. The models are trained sequentially on task sequence Rain800-Rain100H using continual learning methods and tested on Rain100H. The baseline is trained on Rain800 solely.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Training on Rain800-Rain100H</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Rain800)</td>
<td>17.93</td>
<td>0.4868</td>
<td></td>
</tr>
<tr>
<td>PIGWM</td>
<td>24.13</td>
<td>0.7736</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>24.38</td>
<td>0.7847</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Ablation Study for Optimization Objective Function.

4.4 Ablation Study

In this section, we conduct ablation studies to verify the importance of each item in Eqn. (9). It can be seen clearly in Table 5 and Fig. 4 that the dual prompt pools are the key to overcoming catastrophic forgetting. Moreover, feature-level prompts play a more critical role, and the parameter regularization technique is able to further improve our model's performance.

Table 5: Ablation Study for Optimization Objective Function.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Training on Rain800-Rain100H, Testing on Rain800</td>
<td>PSNR</td>
<td>SSIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>24.39</td>
<td>0.8365</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>23.82</td>
<td>0.8262</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>22.67</td>
<td>0.8189</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>23.07</td>
<td>0.8311</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>23.54</td>
<td>0.8215</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper, we propose a novel continual learning scheme for single image deraining, Dual Prompt Learning (DPL). It learns to select and update compact prompts, i.e., small learnable parameters, to instruct the transformer-based rain removal model at both image and feature levels for continual rain removal. As regularization technique is further adopted to penalize the intense changes of important parameters jointly with the learned prompts. Extensive experi-
mentation on various rain streak benchmarks demonstrates the effectiveness of our proposed scheme. Additionally, this approach can be seamlessly integrated into the training of lower-level task models, resulting in improved adaptability and functionality in challenging environments.

Acknowledgements
This work is supported by the Fundamental Research Funds for the Central Universities and the National Natural Science Foundation of China under Contract No.61772043 and a research achievement of Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). This research work is also partially supported by the Basic and Frontier Research Project of PCL and the Major Key Project of PCL.

References


