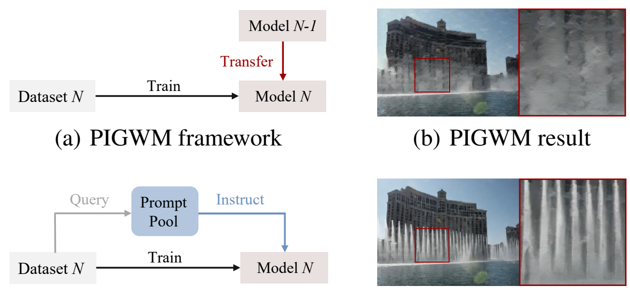
Dual Prompt Learning for Continual Rain Removal from Single Images

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



(a) PIGWM framework (b) PIGWM result (c) DPL framework (d) DPL result

Figure 1: The comparison between the traditional continual learning method and our method. (a) Parameter regularization framework, *e.g.*, PIGWM. [Zhou *et al.*, 2021] (b) Result of PIGWM (PSNR/SSIM: 21.47 dB/0.6792). (c) Overview of our Dual Prompt Learning (DPL) framework. The DPL framework only utilizes a single model and prompt pools to store task-specific knowledge as prompts, *i.e.*, learnable parameters, as the past memory. DPL selectively updates prompts in an instance-wise manner, which are optimized to guide model prediction and explicitly manage task-invariant and task-specific knowledge while preserving model plasticity. (d) Result of DPL (PSNR/SSIM: 27.79 dB/0.8240).

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.

[*]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; Pascanu 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[Pascanu 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 [Pfeiffer *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 prompt 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 $\mathcal{D} = \{D_1, \cdots, D_T\}$ where the t-th task $D_t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ contains n_t pairs of samples, where each pair consists of a rainy image $x_i^t \in \mathcal{X}$ and its clean corresponding background $y_i^t \in \mathcal{Y}$. Our goal is to train a model $f\left(\cdot|\theta_{model}\right) : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by θ_{model}, where $\mathcal{X}$ is the rain images domain and $\mathcal{Y}$ is the corresponding clean images domain. However, the data of $D_1, D_2, \cdots, D_n$ 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_e is the input embedding layer, and f_r 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 $\mathcal{P} = \{P_1, \cdots, P_M\}$ containing a certain amount of trainable prompts $P_i = (K_i, V_i)$ where value $V_i \in \mathbb{R}^{C\times H\times W}$ is a representation with the same size as the input image, and key $K_i = f_e\left(V_i\right)$ is the embedded value to match the key of input. Following the notations in Sec. 3.1, x and $x_e = f_e\left(x\right)$ are the input and its corresponding embedding feature, respectively. Note that we omit the task index t of x 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 γ to measure the similarity of a prompt and the input image. We directly take γ 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 $Q_n = [q_1^n, q_2^n, \cdots, q_M^n]$, where each entry represents the normalized frequency of prompt P_i 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:

$$\mathcal{P}_x = \underset{\{s_i\}_{i=1}^N \subseteq [1,M]}{\arg\min} \sum_{i=1}^{N} \gamma\left(K_{s_i}, f_e\left(x\right)\right) \cdot q_{s_i}^n, \tag{1}$$

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.

$$x_1 = [V_{s_1}; V_{s_2}; \cdots; V_{s_N}; x], \tag{2}$$

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 $x_e = f_e\left(x\right) \in \mathbb{R}^{L\times D}$ 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 $\mathcal{P}' = \{P_1', P_2', \cdots, P_M'\}$ containing a certain amount of prompts $P_i' = (K_i', V_i')$ where $K_i' = V_i' \in \mathbb{R}^{L\times D}$, for $i = 1, 2, \cdots, M$, and its corresponding prompt frequency table for task n, $Q_n' = [{q_1^n}', {q_2^n}', \cdots, {q_M^n}']$, where each entry represents the normalized frequency of prompt P_i' being selected in task n. Similar to Sec. 3.2, we denote γ as a cosine distance function to score the match between $x_e = f_e\left(x\right)$ and prompts $P_1', P_2', \cdots, P_M'$. In other words, during task n, we lookup the top-N prompts by simply solving the following objective:

$$\mathcal{P}_x' = \underset{\{s_i'\}_{i=1}^N \subseteq [1,M]}{\arg\min} \sum_{i=1}^{N} \gamma\left(K_{s_i'}', f_e\left(x\right)\right) \cdot {q_{s_i'}^n}', \tag{3}$$

where $\{s_1', s_2', \cdots, s_N'\}$ is a subset of N indices from $[1, M]$, and we can then adapt the input embedding as follows:

$$x_2 = [V_{s_1'}'; V_{s_2'}'; \cdots; V_{s_N'}'; f_e\left(x_1\right)], 1 \leq N \leq M, \tag{4}$$

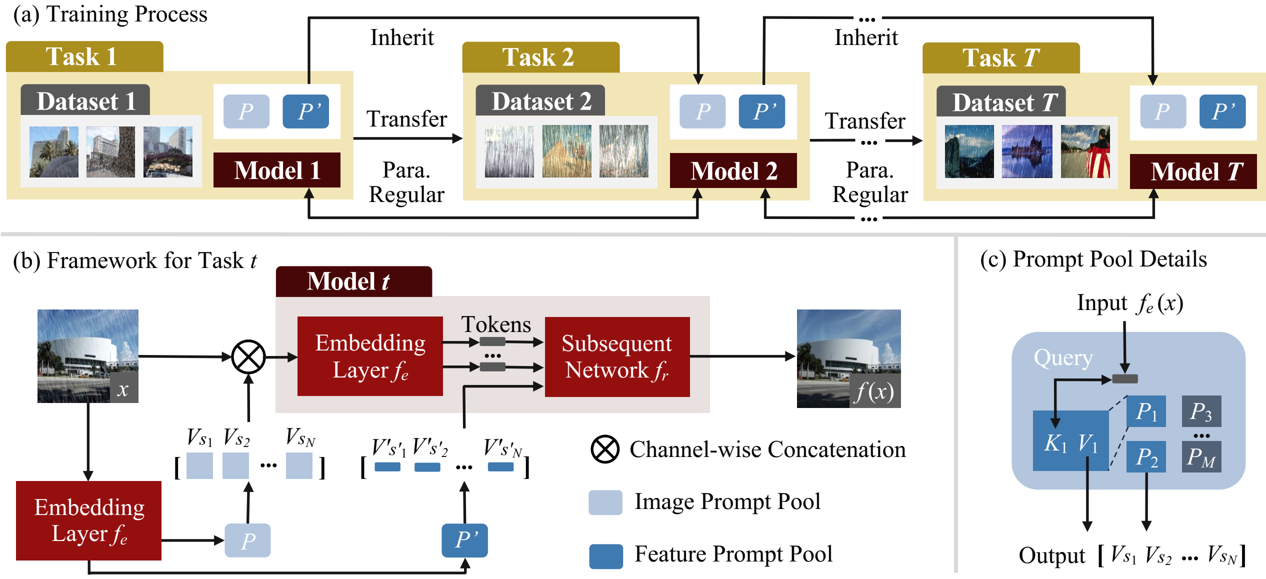


Figure 2: The framework of our proposed dual prompt learning for single image rain removal. (a) The training process. (b) The framework for Task t. (c) Details of the prompt pools. First, the input adaptively selects top-N image prompts by measuring the similarity of a prompt and the input image. After selecting, the top-N prompts and the input are concatenated along the channel dimension and fed into the embedding layer together. Similarly, a subset of prompts from the feature prompt pool based on our proposed instance-wise query mechanism is selected and concatenated with embedded tokens along the token length dimension for further training. The objective is to learn to select and update prompts to instruct the transformer-based rain removal model.

where $[;]$ 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 $x_1 = [V_{s_1}; V_{s_2}; \cdots; V_{s_N}; x]$. 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 $\theta^n_{model} = \{\theta^n_1, \theta^n_2, \cdots, \theta^n_r\}$ where r is the depth of the network, and the parameter set of the overall prompts is denoted as $\theta^n_{prompt} = \{\theta'^n_1, \theta'^n_2, \cdots, \theta'^n_s\}$, where s is the total quantity of all parameters pertained to these prompts. We signify $\theta^n = \theta^n_{model} \cup \theta^n_{prompt}$. For the sake of convenience, we imply $\theta^n = \{\theta^n_1, \theta^n_2, \cdots, \theta^n_{r+s}\}$. X^n and Y^n indicate the rain images set 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:

$$\begin{aligned}\Delta f(\theta^{n+1}, &\theta^n, x, y) = Dist\left(f\left(x, \theta^n\right), f\left(x, \theta^{n+1}\right)\right) \\ &\triangleq \left| l\left(f\left(x, \theta^n\right), y\right) - l\left(f\left(x, \theta^{n+1}\right), y\right)\right|,\end{aligned} \tag{5}$$

where $|\cdot|$ denotes the absolute value operator, l represents the weighted average of the perceptual loss, L_1 loss, and prompt distance loss used for training the de-raining network. For the k-th depth parameter on Task n, θ^n_k, we denote

$$\delta\theta^n_k = \theta^{n+1}_k - \theta^n_k, \tag{6}$$

To evaluate $\Delta f\left(\theta^{n+1}, \theta^n, x, y\right)$, we take the Taylor expansion of $l\left(f\left(x, \theta\right), y\right)$ at point θ^{n+1}_k, which is an infinite sum of terms that are expressed in the form of target functions derivatives at a single point:

$$\begin{aligned} l\left(f\left(x, \theta^{n+1}_k\right), y\right) = {} & l\left(f\left(x, \theta^n_k\right), y\right) + \left(\nabla_{\theta^n_k} l\right)^T \cdot \delta\theta^n_k \\ & + \frac{1}{2}\left(\delta\theta^n_k\right)^T \cdot H \cdot \left(\delta\theta^n_k\right) \\ & + O\left\|\left(\delta\theta^n_k\right)^3\right\|, \end{aligned} \tag{7}$$

where $H = \nabla^2_{\theta^n_k} l\left(\theta^n_k, x\right)$ denotes Hessian matrix for θ^n_k. In the actual calculation, we abandon the items more than three times. Therefore we have:

$$\begin{aligned} & I\left(\theta^{n+1}\right) = \Delta f\left(\theta^{n+1}, \theta^n, x, y\right) \\ & = \sum_{k=1}^{r+s}\left[l\left(f\left(x, \theta^{n+1}_k\right), y\right) - l\left(f\left(x, \theta^n_k\right), y\right)\right] \\ & = \sum_{k=1}^{r+s}\left[\left(\nabla_{\theta^n_k} l\right)^T \cdot \delta\theta^n_k + \frac{1}{2}\left(\delta\theta^n_k\right)^T \cdot H \cdot \left(\delta\theta^n_k\right)\right], \end{aligned} \tag{8}$$

A small $I\left(\theta^{n+1}\right)$ 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:

$$\begin{aligned}\mathcal{L}_x = \min_{\mathcal{P},\mathcal{P}',\theta} & \alpha L_1\left(f\left(x\right),y\right) + \beta L_p\left(f\left(x\right),y\right)\\ &+ \zeta\sum_{\mathcal{P}_x}\gamma\left(f_e\left(P_{s_i}\right), f_e\left(x\right)\right)\cdot q^n_{s_i}\\ &+ \eta\sum_{\mathcal{P}'_x}\gamma\left(P'_{s'_i}, f_e\left(x\right)\right)\cdot {q^n_{s'_i}}' + \omega I\left(\theta^n\right).\end{aligned} \tag{9}$$

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, $\mathcal{P}$ and $\mathcal{P}'$ are initialized randomly. In detail, for a $P_i = (K_i, V_i), 1 \le i \le M$, 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 $P'_i = (K'_i, V'_i), 1 \le i \le M$, 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 $B = \{(x^t_i, y^t_i)\}^l_{i=1}$, where l is the batch size, two sets of chosen prompts $\mathcal{P}_B, \mathcal{P}'_B$ 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 $\mathcal{P}_x$ and top-N feature prompts $\mathcal{P}'_x$ are obtained by solving Eqn. (1) and Eqn. (3). We update the sets of chosen prompts by $\mathcal{P}_B = \mathcal{P}_B \cup \mathcal{P}_x$, $\mathcal{P}'_B = \mathcal{P}'_B \cup \mathcal{P}'_x$, 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 $\mathcal{L}_x$ of every input in the mini-batch by Eqn. (9). We calculate per batch loss $\mathcal{L}_B$ by accumulating $\mathcal{L}_x$, and we update the parameters in $\mathcal{P}$ and $\mathcal{P}'$ by back-propagation immediately. In other words, $\mathcal{P}$ and $\mathcal{P}'$ are updated once for each epoch. The whole process of our proposed DPL for continual rain removal is summarized in Algorithm 1.

Algorithm 1 Dual Prompt Learning

Require: DataSet $\mathcal{D} = \{D_1, D_2, \cdots, D_T\}$,where D_i includes an amount of pairs of rainy/clean images$\{(x_i, y_i)\}^L_{i=1}$ and T is the number of tasks.

Parameter: Network f and overall prompts $\mathcal{P}, \mathcal{P}'$parameterized by θ, image-level prompt pool $\mathcal{P} = \{P_1, \cdots, P_M\}$, feature-level prompt pool $\mathcal{P}' = \{P'_1, P'_2, \cdots, P'_M\}$, where $P_i = (K_i, V_i), P'_i = (K'_i, V'_i)$, for $1 \le i \le M$. Maintained frequency set $Q_t = [q^t_1, q^t_2, \cdots, q^t_M]$, $F'_t = [{q^t_1}', {q^t_2}', \cdots, {q^t_M}']$ for the t-th task, number of training epochs of the t-th task E_t, $1 \le t \le T$, learning rate r, hyperparameters of loss function $\alpha, \beta, \zeta, \eta, \omega$.

Begin

```
Load the pre-trained VGG-16 model, and get D = {D_1, D_2, ..., D_T}.
for t = 1, 2, ..., T do
    Initialize Q_t and Q'_t with real number 1. Initialize P and P'
    with random prompt-sized items.
    for e = 1, 2, ..., E_t do
        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 = {}.
        for (x, y) in B do
            Calculate the embedding input x_e = f_e(x).
            Lookup top-N image prompts by solving Eqn. (1).
            Prepending x_1 with corresponding top-N prompts by
            x_1 = [V_{s_1}; V_{s_2}; ...; V_{s_N}; x].
            Calculate the embedding input f_e(x_1).
            Lookup top-N feature prompts by solving Eqn. (3).
            Prepending x_2 with corresponding top-N prompts by
            x_2 = [V'_{s'_1}; V'_{s'_2}; ...; V'_{s'_N}; f_e(x_1)].
            Calculate the rain-free result f(x) = f_r(x_2).
            Calculate per sample loss by solving Eqn. (9).
            Update sets of chosen prompts: P_B = P_B ∪ P_x.
            Update sets of chosen prompts: P'_B = P'_B ∪ P'_x.
            Update Q_t and Q'_t by adding 1 to the frequency-items
            corresponding to the selected prompts.
        end for
        Calculate per batch loss L_B by accumulating L_x.
        Update P and P'.
    end for
    Update θ.
end for=0
```

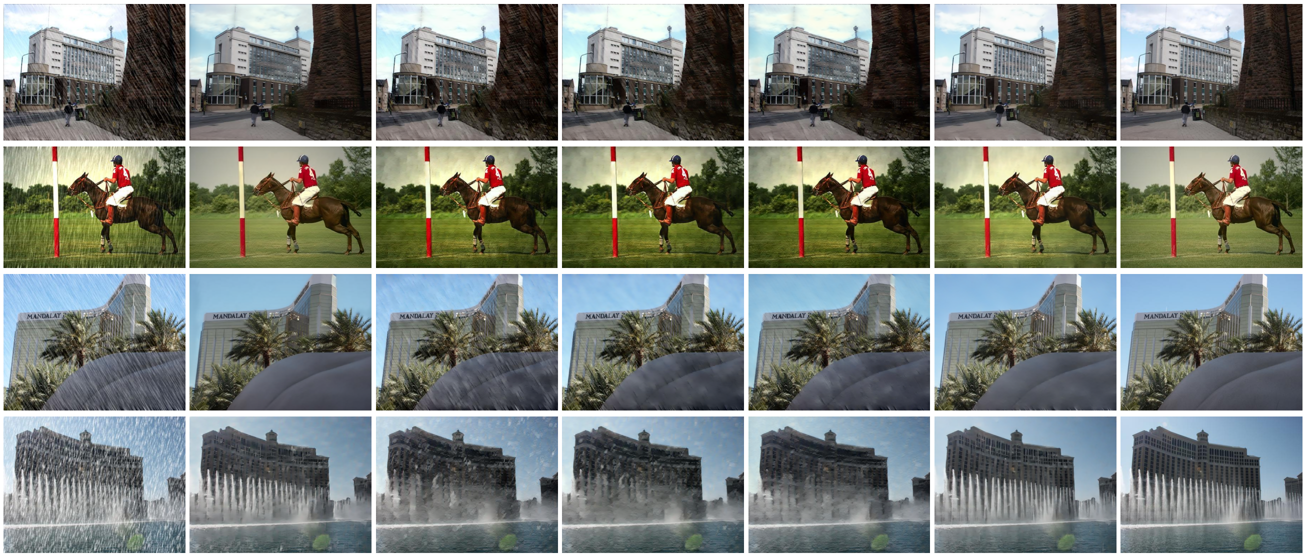
End

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.

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.



(a) Input (b) Task0 (c) SI Task1 (d) Replay Task1 (e) PIGWM Task1 (f) **DPL Task1** (g) GT

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.

	Replay	Parameter Reg. (PR)	Dual Prompt (DP)	DP + PR
Performance (PSNR)	22.76	22.48	22.96	24.39
Increased Parameter Ratio	100% + 4%× tasks	100%	5.2%	105.2%

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

Methods	Training on Rain800-Rain100H		
	PSNR	SSIM	Degradation on Rain800
Baseline (only Rain800)	26.63	0.8583	0, 0
Ours (only Rain800)	**27.52**	**0.8667**	-0.89, -0.0084
SI	19.87	0.6451	6.76, 0.2132
EWC	21.64	0.7962	4.99, 0.0621
Replay	22.76	0.8136	3.87, 0.0447
Deep generative	22.51	0.8162	4.12, 0.0421
PIGWM	22.48	0.8058	4.15, 0.0525
Ours	**24.39**	**0.8365**	**2.24, 0.0218**

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 γ as cosine distance. For both prompt pools, we assign M to 100, and N to 18, since the mini-batch size is 18. In terns of the hyperparameters, we assign α to 1, β to 0.04, ω to 0.95, and ζ, η 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

(a) (b) (c) (d) (e) (f)

Figure 4: Ablation study results of sequential training on Rain800-Rain100H and testing on Rain800. (a) All components are applied; (b) No Image Prompts; (c) No Feature Prompts; (d) No Parameter Regularization; (e) No Prompts Regularization; (f) Input image.

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.

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

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.

Methods	Training on Rain800-Rain100H	
	PSNR	SSIM
Baseline(Rain800)	17.93	0.4868
PIGWM	24.13	0.7736
Ours	**24.38**	**0.7847**

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.

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.

Training on Rain800-Rain100H, Testing on Rain800					
Image Prompts	Feature Prompts	Parameter Reg.	Prompts Reg.	PSNR	SSIM
✓	✓	✓	✓	24.39	0.8365
×	✓	✓	✓	23.82	0.8262
✓	×	✓	✓	22.67	0.8189
✓	✓	×	✓	23.07	0.8311
✓	✓	✓	×	23.54	0.8215

Table 5: Ablation Study for Optimization Objective Function.

4.5 Extension to Multiple Datasets

In this section, we prove that our method shows superior performance not only for two tasks but also for multiple tasks. Namely, for n task sequences, we continually train task n after the model has been trained for the first $n-1$ tasks. Taking three tasks as an example, the model is trained for the first two tasks with our proposed scheme. Then, we continue to train the model for Task 3 based on this model. Table 6 shows the result of a model trained on Rain800-Rain100H-Rain100L sequentially, to verify the effectiveness of our method. The results clearly demonstrate the superiority of our method.

Method	Testing Set		
	Rain800	Rain100H	Rain100L
SI	21.30/0.7152	16.53/0.5946	36.96/0.9800
Replay	22.77/0.7758	18.78/0.6861	32.12/0.9541
PIGWM	22.80/0.7564	17.44/0.6331	32.58/0.9561
DPL	**24.44/0.8163**	**19.63/0.7282**	31.93/0.9589
Reference	26.63/0.8583	28.49/0.8802	37.97/0.9825

Table 6: PSNR/SSIM results trained on sequential tasks Rain800-Rain100H-Rain100L.

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 experimentation 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

[Aljundi *et al.*, 2018] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory Aware Synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[Chen *et al.*, 2018] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[De Lange *et al.*, 2021] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[Fu *et al.*, 2017] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Hu *et al.*, 2021] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv:2106.09685*, 2021.

[Kirkpatrick *et al.*, 2017] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 2017.

[Lester *et al.*, 2021] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv:2104.08691*, 2021.

[Li and Hoiem, 2017] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[Li and Liang, 2021] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv:2101.00190*, 2021.

[Li *et al.*, 2018a] Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. Non-locally enhanced encoder-decoder network for single image de-raining. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2018.

[Li *et al.*, 2018b] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[Liu *et al.*, 2021] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv:2107.13586*, 2021.

[Mai *et al.*, 2022] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 2022.

[Mallya *et al.*, 2018] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[Pascanu and Bengio, 2013] Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. *arXiv:1301.3584*, 2013.

[Pfeiffer *et al.*, 2020] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv:2005.00247*, 2020.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 2020.

[Ren *et al.*, 2019] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Valanarasu *et al.*, 2022] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M. Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[Wang *et al.*, 2004] Zhou 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 (TIP)*, 2004.

[Wang *et al.*, 2020] Lijun Wang, Jianming Zhang, Oliver Wang, Zhe Lin, and Huchuan Lu. SDC-depth: Semantic divide-and-conquer network for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[Wang *et al.*, 2022] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[Xu and Zhu, 2018] Ju Xu and Zhanxing Zhu. Reinforced continual learning. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[Yang *et al.*, 2017] Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Yang *et al.*, 2019] Wenhan Yang, Robby T. Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[Zamir *et al.*, 2021] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[Zenke *et al.*, 2017] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[Zhang and Patel, 2018] He Zhang and Vishal 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 (CVPR)*, 2018.

[Zhang *et al.*, 2019] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. Image de-raining using a conditional generative adversarial network. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2019.

[Zhang *et al.*, 2020] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C.-C. Jay Kuo. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[Zhao *et al.*, 2021] Hanbin Zhao, Hui Wang, Yongjian Fu, Fei Wu, and Xi Li. Memory efficient class-incremental learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2021.

[Zhou *et al.*, 2021] Man Zhou, Jie Xiao, Yifan Chang, Xueyang Fu, Aiping Liu, Jinshan Pan, and Zheng-Jun Zha. Image de-raining via continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.