RESTORING WITHOUT FORGETTING: FILTER-LEVEL CONTINUAL IMAGE RESTORATION VIA PARAMETER-SPACE INTEGRATED GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting image restoration models to a stream of new tasks without revisiting past data remains challenging due to catastrophic forgetting. In this work, we propose Restoring without Forgetting (RwF), a filter-level continual adaptation framework for image restoration built upon a critical observation: task-specific knowledge is centered in a small subset of filters and can be separated from those reconstructing general content. RwF first performs parameter-space integrated gradients attribution to localize degradation-critical filters in a coarse-to-fine manner. It then adapts to new tasks by generating task-specific filters from a filter bank using compact factorized low-rank transformations, further augmented with cross-task attention and prototypical contrastive learning, and lastly assembles them back only at localized positions. Experiments on six restoration tasks show that RwF effectively avoids forgetting and achieves competitive restoration quality against all-in-one methods that have full data access, and outperforms LoRA-style adaptation with $\sim 10\times$ fewer additional parameters.

1 INTRODUCTION

Recent image restoration methods are evolving from the single-task specialist model (Mansour & Heckel, 2023; Chen et al., 2023; Kong et al., 2023; Wang et al., 2023; Gao et al., 2024; Dong et al., 2015) to unified backbones (Jiang et al., 2024; Cui et al., 2025; Potlapalli et al., 2023) that handle multiple degradations simultaneously. Nevertheless, real-world deployments rarely encounter a fixed and closed set of degradations: new corruption types, capture conditions, and device characteristics emerge over time. In many practical scenarios, revisiting historical training data is infeasible due to storage, privacy, or data ownership constraints. As a result, naively fine-tuning a restoration model on each incoming task often leads to catastrophic forgetting (Goodfellow et al., 2013), naturally framing the problem as continual image restoration.

While continual learning has been extensively studied for high-level recognition (Kirkpatrick et al., 2017; Rebuffi et al., 2017; Qu et al., 2021; Joseph et al., 2022; Sun et al., 2023), extending it to image restoration is more challenging. First, restoration is evaluated at the pixel level, where even mild forgetting can cause visible artifacts and sharp drops in terms of metrics. Second, restoration models usually learn degradation patterns implicitly, and these patterns are entangled with image content, thus generic regularization often struggles to preserve task-specific performance. Existing replay-based methods (Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017; Wang et al., 2022a; Shin et al., 2017; Cong et al., 2020) require storing samples from previous tasks, which is often impractical when access to past data is limited, and it also increases training cost over time. Architectural expansion approaches (Rusu et al., 2016; Serra et al., 2018; Hung et al., 2019; Kang et al., 2022) can protect past knowledge but often incur significant parameter growth when applied to modern restoration networks. Parameter-efficient adaptation methods, like Adapter (Houlsby et al., 2019; Chen et al., 2022) or LoRA (Hu et al., 2022), reduce per-task overhead, yet they generally inject trainable parameters broadly and do not explicitly identify which parts of the backbone are truly responsible for modeling each degradation, limiting both scalability and robustness under sequential adaptation.

We address these limitations with Restoring without Forgetting (RwF), a filter-level continual adaptation framework inspired by a critical observation (Xie et al., 2021; Sundararajan et al., 2017a): degradation-specific knowledge is centered in a small subset of filters and can be separated from

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

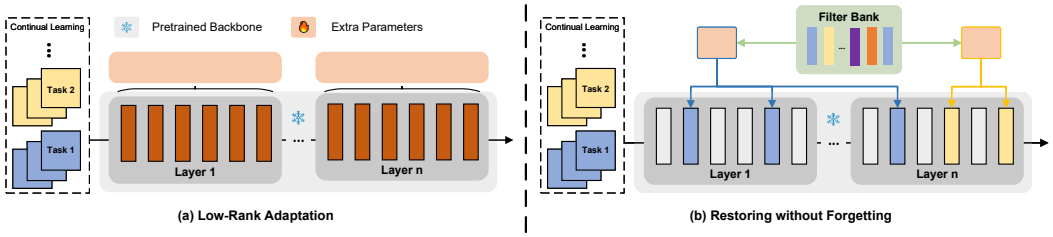

Figure 1: (a) LoRA updates every filter of each adapted layer through low-rank matrices. (b) RwF localizes the few degradation-critical filters of each task and regenerates only them from a shared filter bank, leaving the rest of the backbone frozen.

filters that primarily reconstruct general content (detailed in Appendix B). Concretely, RwF first performs parameter-space integrated gradients attribution, and localizes degradation-critical filters through integrated gradient values in a coarse-to-fine manner. To efficiently acquire new restoration capabilities, RwF maintains a filter bank of previously localized filters and generates task-specific filters using compact factorized low-rank transformations, which are then assembled back only at the localized positions. We further introduce cross-task attention to reweight historical knowledge during filter generation, and perform prototypical contrastive learning to promote task-separated representations. Our contributions are summarized as follows:

- We propose a filter-level continual adaptation framework to achieve Restoring without Forgetting (RwF) for continual image restoration under a sequential setting where past-task data cannot be revisited, effectively avoiding catastrophic forgetting.
- We introduce integrated gradients in parameter space to identify and retain a small set of degradation-critical filters, then generate task-specific filters from a filter bank via factorized low-rank transforms.
- Extensive experiments on six restoration tasks demonstrate that RwF significantly avoids forgetting, achieves competitive quality compared with all-in-one restoration models, and outperforms continual learning methods and LoRA-style adaptation while requiring about an order of magnitude fewer additional parameters.

2 Related Work

Image Restoration has progressed from task-specific models (Mansour & Heckel, 2023; Chen et al., 2023; Kong et al., 2023; Wang et al., 2023; Gao et al., 2024; Dong et al., 2015) to generic backbones (Wang et al., 2022b; Feng et al., 2023; Zamir et al., 2021) and all-in-one models (Potlapalli et al., 2023; Jiang et al., 2024; Cui et al., 2025) that handle multiple degradations at once, yet all assume joint training with every degradation available upfront. When corruptions arrive sequentially in the continual learning setting, fine-tuning on each catastrophically forgets the others.

Continual Learning learns a sequence of tasks without catastrophic forgetting. *Regularization-based* methods restrict updates to parameters important for past tasks (Kirkpatrick et al., 2017; Zeng et al., 2019) or distill previous outputs (Li & Hoiem, 2017) but weaken over long task streams (Hsu et al., 2018; Farquhar & Gal, 2018); *replay-based* methods rehearse stored or generated samples (Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017; Wang et al., 2022a; Shin et al., 2017; Cong et al., 2020) at the cost of computation and privacy; *architecture-based* methods isolate task-specific parameters (Rusu et al., 2016; Serra et al., 2018; Hung et al., 2019; Kang et al., 2022) at the cost of scalability. These methods are mainly designed for classification tasks, where a discrete label absorbs feature drift, but restoration demands pixel-level fidelity, so drift surfaces as visible artifacts, motivating adaptation confined to the parameters each task relies on.

Parameter-Efficient Fine-Tuning methods such as adapters (Houlsby et al., 2019; Chen et al., 2022) and LoRA (Hu et al., 2022) bound per-task overhead with few added parameters, but place them uniformly across layers, blind to which components encode a degradation. Continual generation methods (Zhai et al., 2020; 2021) factorize filters into shared and task-specific parts yet fix the split per layer, and LIRA (Liu et al., 2020) adds a degradation-specific branch per task with distillation and GAN-based pseudo-replay, paying for both expansion and replay. In contrast, we localize the filters each degradation relies on via Integrated Gradients (Sundararajan et al., 2017b;a) and adapt only those, separating content reconstruction from degradation-specific behavior without extra branches or replay.

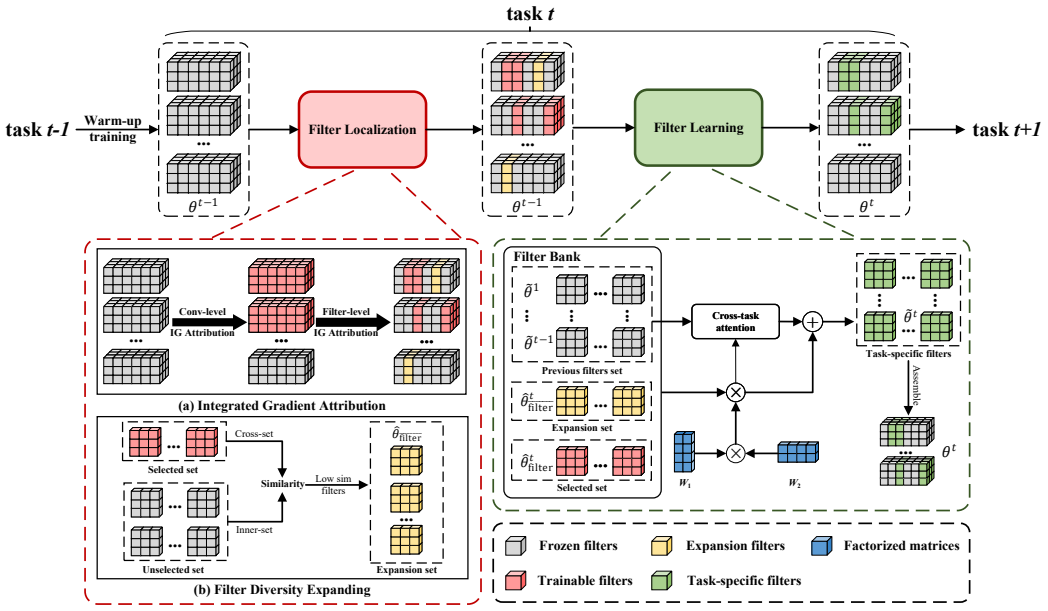

Figure 2: Overview of RwF. Filter localization (left): parameter-space integrated gradients rank layers, then filters within them, and the selected set is expanded with high-IG, low-similarity filters. Filter learning (right): the localized filters are regenerated from a filter bank by factorized matrices with cross-task attention and assembled back into the frozen backbone.

3 METHOD

3.1 OVERVIEW

In this work, we propose an integrated gradient-based filter learning approach to solve the problem of continual learning for image restoration models. The main objective of this problem is to train a model $\mathcal{M}$ parameterized by θ that can handle a series of image restoration tasks $\{\mathcal{T}_t\}_{t=1}^{T}$, constrained by the condition that the learning process can only access one task at a time and training data for each task is available only once.

Formally, an incoming training sample of task t can be represented as $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{Y}_t\}$, herein $\mathcal{X}_t$ is the input degraded images, $\mathcal{Y}_t$ is the corresponding groundtruth, $t \in \mathcal{T} = \{1, ..., T\}$ is the index of image restoration tasks. The objective function of this problem can be summarized as

$$\min_{\theta} \sum_{t=1}^{T} \mathbb{E}_{(\mathcal{X}_t, \mathcal{Y}_t)} \left\| \mathcal{M}(\mathcal{X}_t) - \mathcal{Y}_t \right\|_1 . \tag{1}$$

As presented in Fig. 2, our RwF model mainly consists of two phases, the filter localization phase and filter learning phase. In the filter localization phase, the restoration model $\mathcal{M}$ is warmed up on the new task for 1k iterations and leverages the integrated gradient method to find the key filters that contribute to modeling the degradation patterns. The localized key filters are employed to serve as prior knowledge to learn new degradation patterns for next task of image restoration. In the filter learning phase, the restoration model $\mathcal{M}$ is able to efficiently perform continual learning by maintaining a filter bank and only updating a small number of key filters for new tasks.

3.2 ADAPTIVE FILTER LOCALIZATION FOR TASK-SPECIFIC DEGRADATION

Revisiting the Integrated Gradient (IG). IG (Sundararajan et al., 2017b;a) is proposed to quantify pixel-wise importance and identify features essential for accurate image recognition. Given a classifier $F : \mathbb{R}^n \to [0, 1]$, an input image $x \in \mathbb{R}^n$, and a black baseline image $\bar{x} \in \mathbb{R}^n$, the (vanilla) integrated gradients are defined as the path integral of the gradients along the straight-line path from $\bar{x}$ to x:

$$\mathrm{IG}_i(x) = (x_i - \bar{x}_i) \int_{\alpha=0}^{1} \frac{\partial F\big(\bar{x} + \alpha \times (x - \bar{x})\big)}{\partial x_i} \, \mathrm{d}\alpha, \tag{2}$$

where x_i denotes the value of i-th pixel and $\frac{\partial F}{\partial x}$ is the gradient of F. Thus, the accumulated gradients are able to reflect the importance of each pixel.

Inspired by previous research in image super-resolution (Xie et al., 2021), the integrated gradient has great potential to find key filters across diverse restoration tasks, thus addressing specific degradations by cumulating filter gradients. Imitating the way of integrated gradient for pixels, we first train a baseline network $\overline{F}$ that only learns to reconstruct the input image itself rather than restoring any degradation. In the meanwhile, a target network F is trained to restore a specific task of image restoration. Building upon the contrast between these two networks, we design a two-step filter localization strategy via integrated gradient.

Integrated gradient-based task-specific filters localization. In order to find the key filters that are essential to restore degradation patterns of a specific restoration task, we accumulate the integrated gradients for all filters by the intermediate model between the baseline model $\overline{F}$ parameterized by $\overline{\theta}$ and target restoration model F. The intermediate model is obtained by $\gamma(\alpha) = \alpha\overline{\theta} + (1-\alpha)\theta$. Thus, Equ. 2 can be approximated by discrete states of intermediate models:

$$\mathrm{IG}_i(\theta) \approx \frac{1}{N}(\theta - \overline{\theta})_i \sum_{k=1}^{N} \left[\frac{\partial \mathcal{L}(\gamma(\alpha), x)}{\partial \gamma(\alpha)} \Big|_{\gamma(\alpha), \alpha = k/N} \right], \tag{3}$$

where $N = 50$ is the number of discrete states. Consequently, integrated gradients effectively measure the importance of each filter for a specific restoration task.

Following Equ. 3, we aim to localize the key filters θ_{filter} for the current restoration task t in a coarse-to-fine manner. Specifically, we coarsely obtain conv-level integrated gradients of all filters to localize the key convolutional layers by

$$\hat{\theta}_{\text{conv}} = \mathbb{I}_{\mathrm{IG}(\theta_{\text{conv}}) > P_k}(\theta), \tag{4}$$

where P_k is the percentile, $\mathrm{IG}(\theta_{\text{conv}}) = \sum \mathrm{IG}(\theta_{\text{filter}})$, $\hat{\theta}_{\text{conv}} = \{\hat{\theta}_{\text{filter1}}, \hat{\theta}_{\text{filter2}}, ...\}$ denotes the selected convolutions with higher IG values. This coarse step allows to coarsely locate the convolutional layers that primarily contribute to degradation restoration rather than basic content reconstruction. Subsequently, within these selected layers, we re-evaluate the integrated gradients at the filter level to precisely identify the key filters and pinpoint their exact locations.

$$\hat{\theta}_{\text{filter}}, L = \mathbb{I}_{\mathrm{IG}(\hat{\theta}_{\text{filter}}) > P_k}(\hat{\theta}_{\text{conv}}), \tag{5}$$

where P_k is the percentile, $\hat{\theta}_{\text{filter}}$ denotes the parameters of preliminarily selected filters with higher IG values, $L = \{l_1, l_2, \ldots, l_r\}$ is the corresponding locations of r selected key filters. Note that the selected filters are only a small fraction of the full restoration model. Therefore, to prevent catastrophic forgetting, our approach explicitly retains the identified filters $\hat{\theta}_{\text{filter}}$. Building upon this precise localization for task t, our continual filter learning mechanism repurposes this minimal filter subset to efficiently adapt to new restoration tasks.

Adaptive IG threshold. In filter localization stage described in Equ. 4 and Equ.5, it adopts a percentile P_k to select key filters. However, different restoration tasks exhibit varying levels of complexity, and the distribution of integrated gradient scores can differ substantially across tasks. A fixed threshold may lead to either excessive or insufficient selection of key filters. Thus, we introduce an adaptive thresholding mechanism that dynamically adjusts the percentile based on the statistical properties of the IG score distribution. Specifically, we compute the variance σ^2 and the information entropy $\mathcal{E}$ of IG scores across all filters:

$$\mathcal{E} = -\sum_i \hat{g}_i \log(\hat{g}_i + \epsilon), \tag{6}$$

where $\hat{g}_i$ is the normalized IG score of the i-th filter and ϵ is a small constant. Based on σ^2 and $\mathcal{E}$, the adaptive percentile is determined by:

$$P_k^{\text{adapt}} = \text{clip}\left(P_k^{\text{base}} + \lambda_\sigma \cdot \sigma^2 - \lambda_\mathcal{E} \cdot \mathcal{E},\ P_{\min},\ P_{\max}\right), \tag{7}$$

where P_k^{base} is the base percentile, λ_σ=0.05 and $\lambda_\mathcal{E}$=0.05 are scaling factors, and $[P_{\min}, P_{\max}]$ defines the valid range (e.g., [85, 98]). Intuitively, when the IG scores are highly concentrated (low variance and low entropy), a higher percentile is used to select only the most critical filters; when the scores are dispersed, a lower percentile is adopted to retain more candidate filters for subsequent diversity expanding. This enables our model to cope with degradation tasks of varying complexity in a self-adaptive manner.

Filter diversity expanding. One potential problem is that the selected key filters $\hat{\theta}_{\text{filter}}$ in Equ. 5 might be similar. To relieve it, we design a filter diversity expanding strategy which supplements unselected filters $\hat{\theta}_{\overline{\text{filter}}}$ by Equ. 5 with higher integrated gradients as well as lower similarity with $\hat{\theta}_{\text{filter}}$. Thus, it can provide adequate prior knowledge to strengthen the performance of subsequent filter learning. Specifically, the filter diversity expanding strategy first calculates the cosine similarity between $\hat{\theta}_{\text{filter}}$ and $\hat{\theta}_{\overline{\text{filter}}}$ by

$$\begin{aligned}\mathcal{S}_j &= \frac{1}{|\hat{\theta}_{\text{filter}}|}\sum_i \cos(\hat{\theta}_i, \hat{\theta}_j) + \frac{1}{|\hat{\theta}_{\overline{\text{filter}}}|}\sum_k \cos(\hat{\theta}_j, \hat{\theta}_k) \\ &= \frac{1}{|\hat{\theta}_{\text{filter}}|}\sum_i \frac{\hat{\theta}_i \cdot \hat{\theta}_j}{\|\hat{\theta}_i\|\,\|\hat{\theta}_j\|} + \frac{1}{|\hat{\theta}_{\overline{\text{filter}}}|}\sum_k \frac{\hat{\theta}_j \cdot \hat{\theta}_k}{\|\hat{\theta}_j\|\,\|\hat{\theta}_k\|},\end{aligned} \tag{8}$$

where $\hat{\theta}_i \in \hat{\theta}_{\text{filter}}$, $\hat{\theta}_j$ and $\hat{\theta}_k \in \hat{\theta}_{\overline{\text{filter}}}$, and $\mathcal{S}$ denotes the similarity score. Then, it selects filters with the top-k lowest similarity scores from $\hat{\theta}_{\overline{\text{filter}}}$ and supplements them to generate the final set of key filters θ_{filter} for new task.

3.3 Continual Filter Learning Mechanism

After filter localization, we propose a continual filter learning mechanism to seamlessly adapt restoration model to new tasks (see Fig. 2), which enables restoration model to continuously learn new task t by optimizing the localized key filters, $\hat{\theta}_{\text{filter}}$.

Continual filter learning. Rather than training independent models for individual restoration tasks, our continual filter learning mechanism optimizes only the transformation matrices W to update the filters θ_{filter} for each new task. Furthermore, we maintain a filter bank to archive previously learned key filters, which serve as prior knowledge for the subsequent continual learning process. Prior to the formal training phase, the model must first trace back to its previous state at $\overline{F}$. Notably, this approach incurs tiny parameter overhead while effectively avoiding catastrophic forgetting by preserving task-specific key filters.

Specifically, for a new task t, the filter learning process (illustrated in Fig. 2) is divided into two steps: filter generation and model assembly. Formally, the filter generation process is defined as:

$$\tilde{\theta}^t = \left[\mathcal{R}^{-1}(\mathcal{R}([\tilde{\theta}^1_{\text{filter}}, \tilde{\theta}^2_{\text{filter}}, \ldots, \hat{\theta}^{t-1}_{\text{filter}}, \hat{\theta}^t_{\overline{\text{filter}}}]) \cdot W^t_1 \cdot W^t_2), \hat{\theta}^t_{\text{filter}}\right], \tag{9}$$

where $\mathcal{R}$ and $\mathcal{R}^{-1}$ denote the reshape operation and its inverse, respectively. $\tilde{\theta}^t$ represents the task-specific filters generated for task t. To enhance parameter efficiency and reduce redundancy, transformation matrix W is factorized into two matrices, W^t_1 and W^t_2.

Following the generation of task-specific filters, the model assembly is formulated as:

$$\theta^t = \mathcal{G}(\theta^{t-1}, \tilde{\theta}^t, L^t), \tag{10}$$

where θ^t denotes complete model parameters for task t, $\mathcal{G}$ symbolizes the assembly operator, and L^t indicates the index of key filters localized for task t. More precisely, the assembly is computed via:

$$\theta^t_i = \sum_{j \notin L^t} \delta_{ij}\theta^{t-1}_i + \sum_{1 \le k \le r^t} \delta_{il^t_k}\tilde{\theta}^t_k, \tag{11}$$

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \tag{12}$$

where δ_{ij} is the Kronecker delta, and θ^t_i represents the i-th filter of the updated model. By decoupling task-specific adaptation from the backbone, our continual filter learning mechanism enables the model to accommodate novel restoration tasks with high computational efficiency. Consequently, the proposed framework effectively eliminates catastrophic forgetting while maintaining a minimal parameter overhead throughout the continual learning process.

Cross-task attention for filter generation. While Equ. 9 generates task-specific filters by matrix multiplication over filter bank, it treats all historical filters equally without considering the varying relevance of different prior tasks to the current task. To selectively leverage and transfer the most

relevant prior knowledge during filter generation, we introduce a lightweight cross-task attention mechanism. Specifically, we treat the currently generated parameter $\tilde{\theta}^t_{\text{cur}}$ as the query, and the historical task parameters $\{\hat{\theta}^i_{\text{filter}}\}_{i=1}^{t-1}$ preserved in the filter bank as keys and values. The cross-task attention computes a weighted aggregation of historical knowledge:

$$\text{Att}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V, \tag{13}$$

where $Q = W_Q\tilde{\theta}^t_{\text{cur}}$, $K = W_K[\hat{\theta}^1_{\text{filter}}, \ldots, \hat{\theta}^{t-1}_{\text{filter}}]$, $V = W_V[\hat{\theta}^1_{\text{filter}}, \ldots, \hat{\theta}^{t-1}_{\text{filter}}]$, and W_Q, W_K, W_V are lightweight linear projections with hidden dimension d. The attended historical knowledge is then injected into the current filter through a residual connection:

$$\tilde{\theta}^t = \tilde{\theta}^t_{\text{cur}} + \gamma \cdot \text{Att}(Q, K, V), \tag{14}$$

where γ=0.1 is a scaling factor to ensure the stability of filter generation process.

3.4 Loss Functions

To optimize our proposed RwF model of previously learned restoration capabilities, we employ a composite loss function comprising three components: pixel reconstruction loss, prototypical contrastive loss, and frequency-domain loss.

Pixel reconstruction loss. Following the typical way of pixel-level supervision for image restoration, we employ the ℓ_1 pixel reconstruction loss to minimize the distance between the restored image $\mathcal{M}(\mathcal{X}_t)$ and the corresponding groundtruth $\mathcal{Y}_t$ for task t: $\mathcal{L}_{\text{pixel}} = \|\mathcal{M}(\mathcal{X}_t) - \mathcal{Y}_t\|_1$.

Prototypical contrastive loss. A key challenge in continual learning is to maintain discriminative feature representations across sequentially learned tasks. To this end, we design a prototypical contrastive loss that encourages the feature representations to form distinguishable clusters for different restoration tasks in the latent feature space. Specifically, we maintain a set of prototypes of feature embeddings $\{p_i\}_{i=1}^t$, each of which serves as the representative embedding for a corresponding degradation task i.

For current task t, we extract latent embeddings e from the intermediate layers of restoration model $\mathcal{M}$. Prototypical contrastive loss pulls e towards the current task prototype p_t while pushing it away from all historical task prototypes $\{p_i\}_{i=1}^{t-1}$:

$$\mathcal{L}_{\text{proto}} = -\log \frac{\exp\left(-\|e - p_t\|_2^2/\tau\right)}{\sum_{i=1}^{K}\exp\left(-\|e - p_i\|_2^2/\tau\right)}, \tag{15}$$

where K is the number of existing tasks and τ=0.07 is the temperature coefficient. In this way, restoration model can maintain a discriminative latent feature space and precisely model diverse degradation patterns across sequential training. To maintain the stability of prototypes during continuous training, we update each prototype using an Exponential Moving Average (EMA):

$$p_t \leftarrow \mu \cdot p_t + (1 - \mu) \cdot \bar{e}_t, \tag{16}$$

where μ=0.9 is a momentum coefficient and $\bar{e}_t = \frac{1}{|\mathcal{B}|}\sum_{f\in\mathcal{B}} f$ denotes the mean embeddings computed over the current mini-batch $\mathcal{B}$. Such momentum update mechanism ensures the prototype evolves smoothly, thereby avoiding the instability caused by noisy gradients in individual batches.

Frequency-domain loss. Considering that most image restoration tasks demand high-fidelity recovery of high-frequency details, we further introduce a frequency-domain loss based on the Fast Fourier Transform (FFT). Given the restored image $\hat{\mathcal{Y}}_t = \mathcal{M}(\mathcal{X}_t)$ and its groundtruth $\mathcal{Y}_t$, we compute the FFT loss as:

$$\mathcal{L}_{\text{FFT}} = \left\|\mathcal{F}(\hat{\mathcal{Y}}_t) - \mathcal{F}(\mathcal{Y}_t)\right\|_1 + \lambda_h \left\|\mathcal{H} \odot \left(\mathcal{F}(\hat{\mathcal{Y}}_t) - \mathcal{F}(\mathcal{Y}_t)\right)\right\|_1, \tag{17}$$

where $\mathcal{F}(\cdot)$ denotes the 2D FFT operation, $\mathcal{H}$ is a high-pass filter that emphasizes high-frequency components, and λ_h=1.5 is a weighting factor. The second term enforces additional supervision on the high-frequency spectrum.

The overall loss function for training our RwF model is:

$$\mathcal{L} = \mathcal{L}_{\text{pixel}} + \alpha\mathcal{L}_{\text{proto}} + \beta\mathcal{L}_{\text{FFT}}, \tag{18}$$

where α=0.1 and β=0.01 are hyper-parameters to balance the contributions of different loss terms.

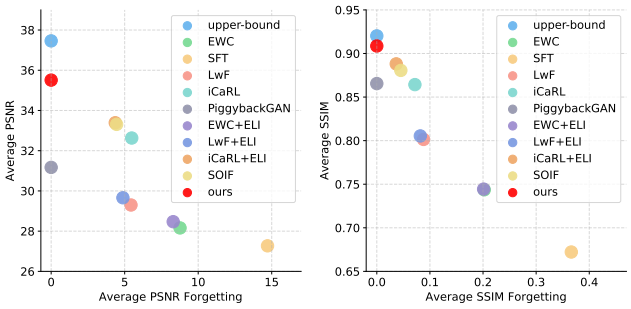

Figure 3: Comparison of forgetting levels for different methods. X-axis stands for the average forgetting, and Y-axis stands for the average metric.

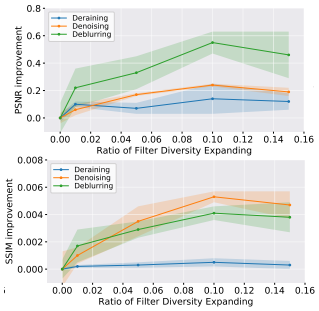

Figure 4: Performance gains with different ratios of filter diversity expanding.

4 EXPERIMENTS

In this section, we first conduct preliminary experiments on self-synthesized datasets with varied degradations to investigate the effectiveness of each RwF property. Then, we evaluate RwF on public restoration benchmarks comprising four diverse restoration tasks with substantially more challenging degradation patterns, and compare against recent parameter-efficient adaptation and all-in-one methods. To demonstrate our method's generality, we equip RwF with three distinct backbones: a **Transformer**-based model HINT (Zhou et al., 2025), a **Convolutional** model MPRNet (Zamir et al., 2021), and a **Diffusion** model DiffUIR (Zheng et al., 2024).

Implementation Details. All experiments employ the AdamW (Loshchilov & Hutter, 2017) optimizer at a constant learning rate, 5×10^{-4} for dehazing and 2×10^{-4} for the remaining stages, with weight decay between 0 and 1×10^{-4} and $(\beta_1, \beta_2) = (0.9, 0.999)$. Each task is trained for 150k iterations with a 1k-iteration warmup. Filter localization uses 50 integration steps over 8 mini-batches with an adaptive percentile threshold between 85 and 88 at tensor level, followed by diversity expansion. The prototypical term maintains one Exponential-Moving-Average (EMA) latent vector per task rather than a sample buffer. We evaluate the quality of restored images in terms of PSNR and SSIM. For more implementation details, please see Appendix C.

4.1 PRELIMINARY EXPERIMENTS

We first conduct preliminary experiments on a synthetic dataset from DIV2K (Agustsson & Timofte, 2017) with diverse degradation patterns, including rain, noise, and blurriness, in order to investigate the effectiveness of our RwF for continual image restoration. Methods are evaluated on all tasks after learning the final task. More experimental details are in Appendix C.

As shown in Tab. 1, our RwF-equipped MPRNet (Zamir et al., 2021) achieves the best restoration performance on the two previously learned tasks against typical continual learning baselines. Compared to PiggybackGAN, our RwF achieves comparable restoration quality (Fig. 3). Qualitative comparisons in Fig. 5 further show better preservation of image structures and textures across tasks.

4.2 ABLATION STUDY

Effect of adaptive task-specific filters localization. Our task-specific filter localization process is decoupled into three progressive stages: Integrated Gradient Attribution (IGA), Convolution-level Localization (CL), and Filter-level Localization (FL). Note that the baseline is LoRA finetuning with the same parameter amount, disabling all our proposed components. As shown in Tab. 2, integrating either CL or FL yields substantial performance gains in continual image restoration over the variant lacking IG attribution. This aligns with the intuition that integrated gradients effectively capture the relative importance of trainable parameters, providing an essential foundation for accommodating new tasks. Moreover, FL consistently outperforms its conv-level counterpart CL, further demonstrating the effectiveness of RwF's coarse-to-fine localization strategy.

Effect of continual filter learning mechanism. We conduct experiments to measure the contribution of Filter Bank (FB), Cross-Task Attention (CTA), and $\mathcal{L}_{\text{proto}}$. Disabling FB and CTA limits selective

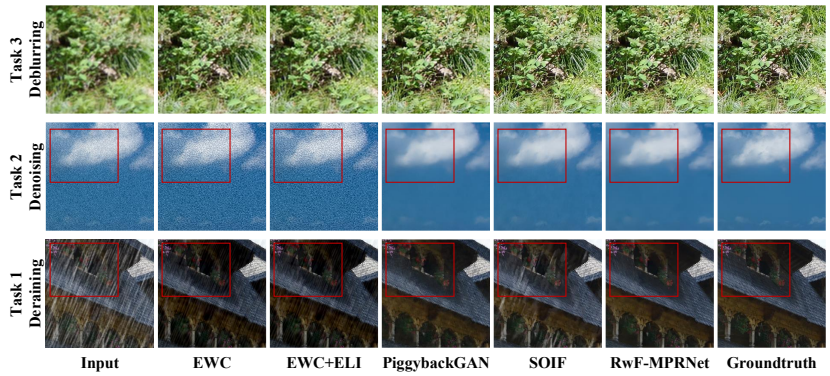

Figure 5: Visualization of restored images on three randomly selected samples from test set. Bounding boxes emphasize remaining degradation artifacts resulting from catastrophic forgetting.

Table 1: Quantitative results after continual learning on three tasks, trained in the order of deraining, denoising, and deblurring.

Method	Deraining		Denoising		Deblurring		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SFT-MPRNet (Zamir et al., 2021)	23.16	0.648	21.48	0.437	**37.16**	**0.922**	27.27	0.670
EWC (Kirkpatrick et al., 2017)	27.33	0.754	26.88	0.642	30.27	0.835	28.16	0.744
LwF (Li & Hoiem, 2017)	29.60	0.853	29.24	0.755	29.05	0.796	29.30	0.801
iCaRL (Rebuffi et al., 2017)	28.88	0.807	32.60	0.865	36.40	0.922	32.63	0.864
PiggybackGAN (Zhai et al., 2020)	32.51	0.895	29.54	0.824	31.46	0.878	31.17	0.866
EWC+ELI (Joseph et al., 2022)	28.07	0.756	27.05	0.643	30.30	0.835	28.47	0.744
LwF+ELI	30.12	0.864	29.80	0.756	29.07	0.796	29.66	0.805
iCaRL+ELI	30.26	0.845	33.48	0.897	36.44	0.922	33.39	0.888
SOIF (Sun et al., 2023)	30.40	0.846	33.10	0.875	36.42	0.921	33.31	0.881
RwF-MPRNet (ours)	**38.52**	**0.942**	**33.69**	**0.874**	34.49	0.913	**35.57**	**0.910**

Table 2: Ablation study on RwF components.

Variant	Average	
	PSNR	SSIM
Baseline	34.64	0.896
w/o CL	35.32	0.906
w/o FL	35.18	0.904
w/o FB	35.12	0.901
w/o CTA	35.26	0.905
w/o $\mathcal{L}_{\text{pro}}$	35.40	0.907
RwF (full)	**35.57**	**0.910**

knowledge transfer, degrading the model's adaptation to new tasks due to interference from irrelevant historical filters. Furthermore, removing $\mathcal{L}_{proto}$ degrades average performance since filter-level retention alone is insufficient without maintaining task-discriminative latent representations. As further shown in Fig. 4, setting the filter diversity expanding ratio to 10% provides an optimal trade-off between performance gain and parameter overhead.

4.3 Experiments on Public Restoration Benchmarks

To further investigate the potential of RwF for continual image restoration, we conduct experiments on a series of restoration tasks with more challenging and diverse degradation patterns, using both public synthetic and real-world restoration datasets.

Experimental Setup. We assemble a four-task benchmark from widely used public datasets: Rain100L (Yang et al., 2017) (deraining), SOTS (Li et al., 2018) (dehazing), Snow100K (Liu et al., 2018) (desnowing), and LOLv2-Real (Yang et al., 2021) (low-light enhancement). Their degradation patterns vary across heterogeneous image conditions and exhibit domain-specific distributions. Besides, we further combine our RwF with a Transformer-based restoration backbone HINT (Zhou et al., 2025) to assess architecture generality.

Baselines. Besides the Sequential Fine-Tuning (SFT) reference, we introduce two additional categories of methods: (1) *all-in-one restoration methods*: including AdaIR (Cui et al., 2025), and AutoDIR (Jiang et al., 2024), which train a single model jointly on all tasks with simultaneous data access; (2) *parameter-efficient fine-tuning*: represented by Low-Rank Adaptation (LoRA) (Hu et al., 2022), which freezes the backbone and learns task-specific low-rank adapters for each new task.

Comparison with all-in-one methods. All-in-one methods access all task data simultaneously, thus their performance should inherently represent the upper bound of continual learning methods. However, we observe that our RwF-HINT model outperforms the all-in-one baseline AutoDIR (Jiang et al., 2024) across four restoration tasks by 1.30 dB average PSNR and approaches the performance of the state-of-the-art all-in-one restoration baseline AdaIR (Cui et al., 2025). It proves that our RwF framework retains the advantages of sequential training and knowledge assembling.

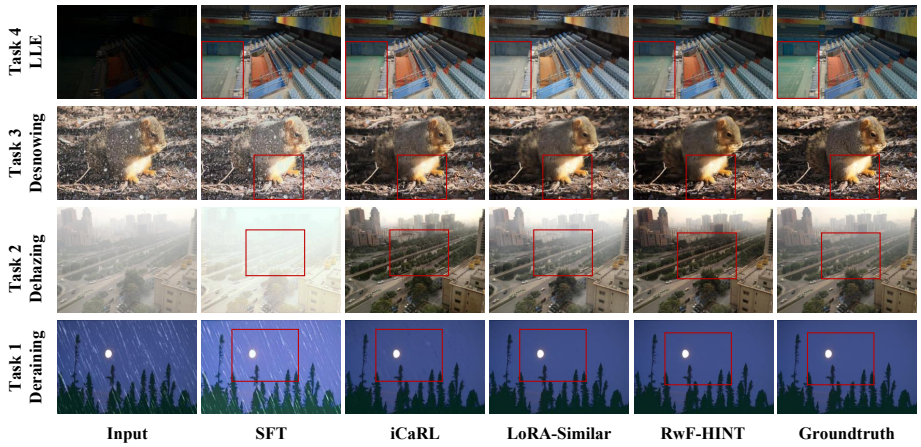

Figure 6: Visual comparison on the four-task public benchmark after learning the last task (LLE).

Table 3: Quantitative results on the four-task public restoration benchmark with the order of deraining, dehazing, desnowing, and low-light enhancement (LLE). Best results are in **bold** and the second are underlined. Note that all-in-one methods are not ranked.

Method	Deraining		Dehazing		Desnowing		LLE		Average		Param
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	increase
AutoDIR (Jiang et al., 2024)	34.49	0.961	29.31	0.970	27.56	0.868	20.54	0.799	27.98	0.900	-
AdaIR (Cui et al., 2025)	36.88	0.980	30.34	0.975	28.95	0.901	21.80	0.821	29.49	0.919	-
SFT-MPRNet (Zamir et al., 2021)	17.29	0.767	13.89	0.712	15.24	0.698	22.40	0.864	17.21	0.760	-
SFT-DiffUIR (Zheng et al., 2024)	23.45	0.819	27.39	0.940	21.20	0.746	21.37	0.860	23.35	0.841	-
SFT-HINT (Zhou et al., 2025)	14.84	0.736	10.33	0.674	11.75	0.574	20.04	0.855	14.24	0.711	-
LwF (Li & Hoiem, 2017)	27.52	0.885	19.84	0.841	20.33	0.782	22.15	0.851	22.46	0.840	-
iCaRL (Rebuffi et al., 2017)	33.11	0.950	26.34	0.946	25.89	0.840	22.68	0.856	27.01	0.898	-
LoRA-Similar (Hu et al., 2022)	**38.72**	**0.984**	27.35	0.958	26.34	0.858	18.13	0.804	27.64	0.901	0.26M
LoRA-Full (Hu et al., 2022)	**38.72**	**0.984**	29.69	0.973	27.97	**0.878**	20.16	0.848	29.14	0.921	1.73M
RwF-MPRNet	36.98	0.979	29.50	0.967	27.48	0.865	20.04	0.838	28.50	0.912	0.59M
RwF-DiffUIR	34.65	0.966	**33.56**	**0.982**	**28.15**	0.872	**23.50**	**0.867**	**29.97**	**0.922**	0.53M
RwF-HINT	**38.72**	**0.984**	29.73	0.973	27.94	0.876	20.73	0.849	29.28	0.921	**0.17M**

Comparison with LoRA. While LoRA (Hu et al., 2022) effectively limits forgetting by constraining adaptation to low-rank subspaces, it has to balance content reconstruction and degradation restoration to train a new task, which obviously limits its performance of continual image restoration: LoRA-Similar trails RwF by 1.64 dB average PSNR on RwF-HINT and LoRA-Full with ten times params than RwF-HINT by 0.14 dB. By contrast, RwF explicitly models task-specific filters for each new restoration with the guidance of previous knowledge, resulting in competitive performance of continual restoration with limited parameter increase.

Qualitative Analysis. Fig. 6 presents visualized results by competing methods across all four restoration tasks. Typical continual learning baselines, SFT and iCaRL (Rebuffi et al., 2017), produce poor results on learned results due to severe catastrophic forgetting. While both LoRA-Similar and our proposed RwF-HINT effectively mitigate forgetting and preserve restoration performance on previously learned tasks, LoRA (Hu et al., 2022) still exhibits visible artifacts and a noticeable loss of high-frequency details. This degradation stems from its insufficient utilization of learned knowledge from prior tasks. In contrast, RwF consistently maintains high-fidelity, task-specific restoration quality across all tasks, avoiding visible artifacts and over-smoothing issues.

5 Conclusion

In this paper, we present Restoring without Forgetting (RwF), a highly efficient continual learning framework tailored for image restoration. By localizing task-critical filters via adaptive integrated gradients, RwF effectively decouples degradation-specific patterns from general content reconstruction. During sequential adaptation, our method dynamically generates and assembles specialized filters using factorized low-rank transformations guided by a historical filter bank. Extensive evaluations over six restoration tasks across three backbones confirm that RwF optimally mitigates catastrophic forgetting, matching the performance of all-in-one restoration models with only a fraction of the parameter overhead.

AI USE STATEMENT

In this work, we used generative AI tools only to check grammar and correct typographical errors in the manuscript text. We did not use generative AI tools to propose the research idea, design the method, implement or run experiments, analyze results, produce figures, or write any of the scientific content. We reviewed every AI-suggested correction before accepting it and take full responsibility for the final content of this work.

REFERENCES

Eirikur Agustsson and Radu Timofte. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *CVPRW*, July 2017.

Haoyu Chen, Jinjin Gu, Yihao Liu, Salma Abdel Magid, Chao Dong, Qiong Wang, Hanspeter Pfister, and Lei Zhu. Masked image training for generalizable deep image denoising. In *CVPR*, pp. 1692–1703, 2023.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. GAN Memory with No Forgetting. *NeurIPS*, 33:16481–16494, 2020.

Yuning Cui, Syed Waqas Zamir, Salman Khan, Alois Knoll, Mubarak Shah, and Fahad Shahbaz Khan. Adair: Adaptive all-in-one image restoration via frequency mining and modulation. In *ICLR*, pp. 57335–57356. International Conference on Learning Representations, ICLR, 2025.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE TPAMI*, 44(5):2567–2581, 2022.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2015.

Sebastian Farquhar and Yarin Gal. Towards Robust Evaluations of Continual Learning. *arXiv preprint arXiv:1805.09733*, 2018.

Xin Feng, Haobo Ji, Wenjie Pei, Jinxing Li, Guangming Lu, and David Zhang. U^2-former: Nested u-shaped transformer for image restoration via multi-view contrastive learning. *IEEE TCSVT*, 34 (1):168–181, 2023.

Ning Gao, Xingyu Jiang, Xiuhui Zhang, and Yue Deng. Efficient frequency-domain image deraining with contrastive regularization. In *ECCV*, pp. 240–257. Springer, 2024.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. *arXiv preprint arXiv:1312.6211*, 2013.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, pp. 2790–2799. PMLR, 2019.

Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines. *NeurIPS Continual Learning Workshop*, 2018.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, Picking and Growing for Unforgetting Continual Learning. *NeurIPS*, 32, 2019.

Yitong Jiang, Zhaoyang Zhang, Tianfan Xue, and Jinwei Gu. Autodir: Automatic all-in-one image restoration with latent diffusion. In *ECCV*, pp. 340–359. Springer, 2024.

KJ Joseph, Salman Khan, Fahad Shahbaz Khan, Rao Muhammad Anwer, and Vineeth N Balasubramanian. Energy-based latent aligner for incremental learning. In *CVPR*, pp. 7452–7461, 2022.

Haeyong Kang, Rusty John Lloyd Mina, Sultan Rizky Hikmawan Madjid, Jaehong Yoon, Mark Hasegawa-Johnson, Sung Ju Hwang, and Chang D Yoo. Forget-free Continual Learning with Winning Subnetworks. In *ICML*, pp. 10734–10750. PMLR, 2022.

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*, 114 (13):3521–3526, 2017.

Lingshun Kong, Jiangxin Dong, Jianjun Ge, Mingqiang Li, and Jinshan Pan. Efficient frequency domain-based transformers for high-quality image deblurring. In *CVPR*, pp. 5886–5895, 2023.

Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE TIP*, 28(1):492–505, 2018.

Zhizhong Li and Derek Hoiem. Learning without Forgetting. *IEEE TPAMI*, 40(12):2935–2947, 2017.

Jianzhao Liu, Jianxin Lin, Xin Li, Wei Zhou, Sen Liu, and Zhibo Chen. LIRA: Lifelong Image Restoration from Unknown Blended Distortions. In *ECCV*, pp. 616–632. Springer, 2020.

Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE TIP*, 27(6):3064–3073, 2018.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient Episodic Memory for Continual Learning. *NeurIPS*, 30, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Youssef Mansour and Reinhard Heckel. Zero-shot noise2noise: Efficient image denoising without any data. In *CVPR*, pp. 14018–14027, 2023.

Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. *NeurIPS*, 36:71275–71293, 2023.

Haoxuan Qu, Hossein Rahmani, Li Xu, Bryan Williams, and Jun Liu. Recent Advances of Continual Learning in Computer Vision: An Overview. *arXiv preprint arXiv:2109.11369*, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental Classifier and Representation Learning. In *CVPR*, pp. 2001–2010, 2017.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*, 2016.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming Catastrophic Forgetting with Hard Attention to the Task. In *ICML*, pp. 4548–4557. PMLR, 2018.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual Learning with Deep Generative Replay. *NeurIPS*, 30, 2017.

Zhicheng Sun, Yadong Mu, and Gang Hua. Regularizing Second-Order Influences for Continual Learning. *CVPR*, 2023.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Gradients of Counterfactuals. *ICLR*, 2017a.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *ICML*, pp. 3319–3328. PMLR, 2017b.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Qiong Wang, Kui Jiang, Zheng Wang, Wenqi Ren, Jianhui Zhang, and Chia-Wen Lin. Multi-scale fusion and decomposition network for single image deraining. *IEEE TIP*, 33:191–204, 2023.

Zhen Wang, Liu Liu, Yiqun Duan, and Dacheng Tao. Continual Learning through Retrieval and Imagination. In *AAAI*, volume 36, pp. 8594–8602, 2022a.

Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A General U-Shaped Transformer for Image Restoration. In *CVPR*, pp. 17683–17693, 2022b.

Liangbin Xie, Xintao Wang, Chao Dong, Zhongang Qi, and Ying Shan. Finding Discriminative Filters for Specific Degradations in Blind Super-Resolution. *NeurIPS*, 34:51–61, 2021.

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 *CVPR*, pp. 1357–1366, 2017.

Wenhan Yang, Wenjing Wang, Haofeng Huang, Shiqi Wang, and Jiaying Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE TIP*, 30:2072–2086, 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 *CVPR*, pp. 14821–14831, 2021.

Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Mach. Intell.*, 1(8):364–372, 2019.

Mengyao Zhai, Lei Chen, Jiawei He, Megha Nawhal, Frederick Tung, and Greg Mori. Piggyback GAN: Efficient Lifelong Learning for Image Conditioned Generation. In *ECCV*, pp. 397–413. Springer, 2020.

Mengyao Zhai, Lei Chen, and Greg Mori. Hyper-LifelongGAN: Scalable Lifelong Learning for Image Conditioned Generation. In *CVPR*, pp. 2246–2255, 2021.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Dian Zheng, Xiao-Ming Wu, Shuzhou Yang, Jian Zhang, Jian-Fang Hu, and Wei-Shi Zheng. Selective hourglass mapping for universal image restoration based on diffusion model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25445–25455. IEEE, 2024.

Shihao Zhou, Dayu Li, Jinshan Pan, Juncheng Zhou, Jinglei Shi, and Jufeng Yang. Devil is in the uniformity: Exploring diverse learners within transformer for image restoration. In *ICCV*, pp. 12307–12317, 2025.

A PRELIMINARY

Inspired by Integrated Gradients (IG) (Sundararajan et al., 2017b;a), our RwF adopts a path-integral view of attribution to localize task-specific filters in image restoration models. Since our continual restoration framework ultimately needs to decide which model components should be preserved and which should be adapted, input-space saliency alone is insufficient. Therefore, we first revisit the principle of IG in its original input-space form, and then explain how the same idea can be migrated to the filter level in parameter space.

A.1 INTEGRATED GRADIENTS IN INPUT SPACE

IG (Sundararajan et al., 2017b;a) was originally introduced as an axiomatic attribution framework for deep networks. Consider a network $F : \mathbb{R}^d \to \mathbb{R}$, an input $x \in \mathbb{R}^d$, and a baseline input $\bar{x} \in \mathbb{R}^d$. The baseline represents the absence of informative evidence, e.g., a black image. Instead of attributing the prediction of $F(x)$ using only the local gradient at x, IG accumulates gradients along the straight-line path from $\bar{x}$ to x:

$$\mathrm{IG}_i(x) = (x_i - \bar{x}_i) \int_0^1 \frac{\partial F(\bar{x} + \alpha \times (x - \bar{x}))}{\partial x_i}\, d\alpha.$$

A key property of IG is the completeness relation, which follows from the chain rule and the fundamental theorem of calculus. Denoting the interpolation path by $\gamma(\alpha) = \bar{x} + \alpha(x - \bar{x})$, we have

$$\sum_{i=1}^{d} \mathrm{IG}_i(x) = \int_0^1 \nabla F(\gamma(\alpha))^\top \frac{\partial \gamma(\alpha)}{\partial \alpha}\, d\alpha = F(x) - F(\bar{x}).$$

IG decomposes the total output difference between the target input and the baseline input into additive contributions from individual input dimensions. In other words, attribution is grounded in a path integral of the model's differential response. In this way, IG attribution is able to identify which input pixels are responsible for the transition from an uninformative baseline input to the final prediction.

Nevertheless, while input-space IG attribution can identify important pixels, it fails to locate task-specific parameters essential for continual image restoration. Since image restoration relies on filters to model complex degradation patterns (e.g., rain or haze), spatial saliency alone cannot pinpoint which model filters should be updated. Furthermore, to ensure selective plasticity and avoid catastrophic forgetting, we shift the attribution target from pixels to filters, thus enabling precise localization and adaptation of task-critical filters during continual learning.

A.2 FROM INPUT-SPACE TO PARAMETER-SPACE ATTRIBUTION

In this section, we provide a derivation of the Equ.3 in Sec.3.2 of the main paper. Let θ be the parameter of the restoration model where $\theta \in \Theta$, $\Theta \subseteq \mathbb{R}^m$ is the whole parameter space, $x \in X$ be the input images where $X \subseteq \mathbb{R}^n$. Thus, the loss function $\mathcal{L} : \Theta \times X \to \mathbb{R}$ can be denoted as $\mathcal{L}(\theta, x)$. For an input image x, we can represent the changes of the model functions by the changes of the loss function:

$$\Delta\mathcal{L} = \mathcal{L}(\theta, x) - \mathcal{L}(\overline{\theta}, x) = \sum_i \int_C \frac{\partial \mathcal{L}(\theta, x)}{\partial \theta_i} d\theta_i, \tag{19}$$

where the $\overline{\theta}$ and θ are the parameters of the baseline model and the target model, respectively. $C \subseteq \Theta$ is the path between $\overline{\theta}$ and θ in parameter space, $\theta \in C$ are all points on C and θ_i is the i-th dimension parameter of θ. Let $\gamma(\alpha)$, $\alpha \in [0, 1]$ be the parameter form of C which satisfies $\gamma(0) = \theta$, $\gamma(1) = \overline{\theta}$. Then, Equ. 19 can be rewritten as:

$$\Delta\mathcal{L} = \mathcal{L}(\gamma(1), x) - \mathcal{L}(\gamma(0), x) = \sum_i \int_0^1 \frac{\partial \mathcal{L}(\gamma(\alpha), x)}{\partial \gamma(\alpha)_i} \times \frac{\partial \gamma(\alpha)_i}{\partial \alpha} d\alpha. \tag{20}$$

Thus, the integrated gradients of the i-th dimension parameter could be defined as:

$$\mathrm{IG}_i(\theta) = \int_0^1 \frac{\partial \mathcal{L}(\gamma(\alpha), x)}{\partial \gamma(\alpha)_i} \times \frac{\partial \gamma(\alpha)_i}{\partial \alpha} d\alpha. \tag{21}$$

A simple and effective choice for the path is the straight line between $\overline{\theta}$ and θ. Then, the path could be represented as $\gamma(\alpha) = \alpha\overline{\theta} + (1-\alpha)\theta$, where $\gamma(0) = \theta$, $\gamma(1) = \overline{\theta}$. Thus, the Equ. 21 can be rewritten as:

$$\mathrm{IG}_i(\theta) = (\theta - \overline{\theta})_i \int_0^1 \frac{\partial \mathcal{L}(\gamma(\alpha), x)}{\partial \gamma(\alpha)_i} d\alpha. \tag{22}$$

Then, the integral can be approximated by discrete states sampled along the path:

$$\mathrm{IG}_i(\theta) \approx \frac{1}{N}(\theta - \overline{\theta})_i \sum_{k=1}^{N} \frac{\partial \mathcal{L}(\gamma(\alpha), x)}{\partial \gamma(\alpha)_i}\bigg|_{\alpha = k/N} \tag{23}$$

B Localized Filters Carry the Restoration Capability

A key observation of our work is that the knowledge a backbone acquires for a degradation is concentrated in a small set of filters. To verify it, we take a trained per-task model, reset a subset of its filters to the baseline $\bar{\theta}$ used for localization, and re-evaluate it in Tab. 4 and Fig. 7.

Resetting the localized filters removes the restoration capability. The localized filters of each task cover only 2.05%–2.89% of the backbone's filters. Resetting them reduces PSNR by 12.18 dB on dehazing, 9.01 dB on desnowing and 11.54 dB on LLIE, which removes 99.5%–109.5% of the gain of each task and brings the model to the level of the degraded input. The result is consistent across six independent localizations of LLIE (11.62 ± 1.10 dB). Moreover, the output becomes almost identical to the degraded input (e.g., 44.51 dB PSNR against the hazy input, compared with 15.88 dB for the intact model), i.e., the model stops restoring rather than producing artifacts.

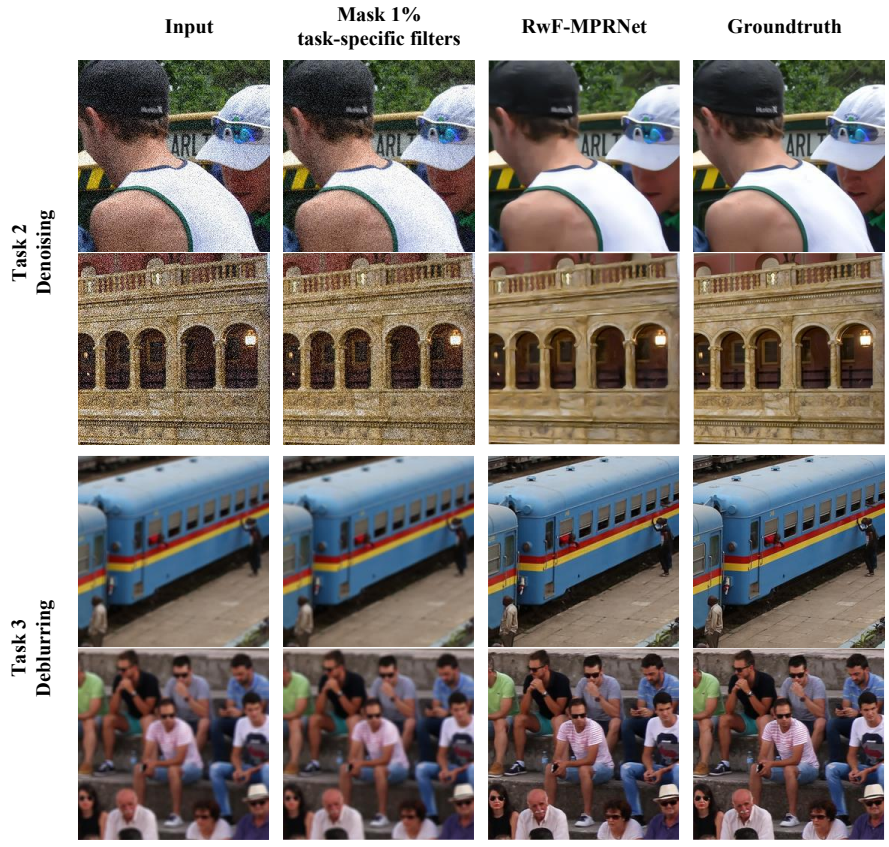

Figure 7: Restored images after resetting 1% of the backbone's filters, localized for each task, to the shared baseline.

Table 4: PSNR (dB) after resetting a fraction of the backbone's filters to the shared baseline. "Localized" resets filters localized for the task and "Random" resets the same number of randomly selected filters (evaluated on 250/150/100 test images).

Filters reset	Dehazing		Desnowing		LLE	
	Localized	Random	Localized	Random	Localized	Random
None (intact)	27.52		27.85		20.23	
0.10%	23.15	26.95	26.34	27.75	19.29	20.76
0.25%	21.15	24.01	24.08	27.47	19.71	21.31
0.50%	19.40	23.72	21.20	27.23	17.13	21.83
1.00%	16.10	18.79	20.19	23.73	13.16	18.01
1.50%	15.85	18.11	20.42	20.67	11.62	16.89
All localized	15.48	18.74	19.00	20.72	8.72	15.47
Degraded input	15.55		18.94		9.72	

Only 1% of the filters is enough. As shown in Tab. 4, resetting only 1% of the filters already reduces PSNR by 11.4 dB on dehazing, 7.7 dB on desnowing and 7.1 dB on LLIE, while resetting the same number of random filters costs much less, e.g., 0.6 dB against 4.4 dB at 0.1% on dehazing. In Fig. 7, we randomly select 1% of the generated filters and replace them with the corresponding filters from the baseline. For both new tasks, when 1% of the generated task-specific filters were masked, the network loses its functionality on the respective new tasks. This observation emphasizes the critical effectiveness of these generated task-specific filters in restoring the corresponding degradation.

The remaining filters stay unchanged. Conversely, all convolution weights outside the localized filters stay bit-exact over the whole task sequence (94.8% of the backbone), which is why RwF does not forget. Note that these experiments show that a task relies on few filters, not that different tasks use disjoint filters: resetting the filters localized for another task can also degrade a task.

C Implementation Details

Benchmark settings. We construct a lightweight benchmark from DIV2K (Agustsson & Timofte, 2017), containing 800 training, 100 validation, and 100 test image pairs. We synthesize three degradation types: rain streaks, Gaussian noise with $\sigma \in \{15, 25, 50\}$, and Gaussian blur with a 25×25 kernel and $\sigma = 1.6$. To create rain streaks, we generate random noise of the same size as the image and retain the noise within a randomized intensity range, apply an affine transformation to obtain a motion kernel with an angle in $[-45°, 45°]$ and a length of 50, blur the kernel with a Gaussian to obtain streak widths of 1 to 5, and convolve the noise with the kernel. For the five-task extension in Tab. 7, haze is synthesized with scattering coefficients in $[0.05, 0.2]$ and global atmospheric light in $[0.9, 1]$, and low light with gamma adjustments in $[1.5, 2]$. RwF is combined with MPRNet throughout these experiments.

Baselines. We compare against sequential fine-tuning (SFT), which uses no anti-forgetting mechanism; regularization-based methods EWC (Kirkpatrick et al., 2017), LwF (Li & Hoiem, 2017), and ELI (Joseph et al., 2022); replay-based methods iCaRL (Rebuffi et al., 2017) and SOIF (Sun et al., 2023); and PiggybackGAN (Zhai et al., 2020), a continual conditional image generation method. We additionally evaluate ELI combined with EWC, LwF, and iCaRL.

Average forgetting. Fig. 3 reports average forgetting (Qu et al., 2021) over previously learned tasks:

$$F_q = \frac{1}{q-1} \sum_{p=1}^{q-1} f_p^q,$$

where q is the number of learned tasks and f_p^q is the difference between the highest performance achieved on task p during continual learning and its performance after learning q tasks. Lower F_q indicates less forgetting.

Warm-up. For each new task ($t \geq 2$), the whole backbone is first trained for 1k iterations. During the first 500 iterations, the model is trained with ℓ_1 and FFT losses while the prototype of the current

Algorithm 1 Pseudocode of RwF in a PyTorch-like style.

```
# f(theta, x) -> (y_hat, e): backbone, outputs the restored image and an embedding
# theta0: pre-trained model of task 1; D[t] = (X[t], Y[t]): data of task t
# L: localized filters; B: filter bank, one filter per row; P[t]: prototype of task t
# W1, W2, w: trainable weights, re-initialized for each task
# L, B, W1, W2 and w are kept per layer; the layer index is omitted

theta_bar = finetune(theta0, X[1], X[1]) # IG baseline (self-reconstruction)
theta, B = theta0, []
P[1] = ema(emb(theta0, X[1]))

for t in range(2, T + 1):
   theta_warm, P[t] = warmup(theta, D[t]) # full fine-tuning (Sec. 3.1)

   # filter localization with integrated gradients (Sec. 3.2)
   ig = (theta_warm - theta_bar) * mean_grad(theta_bar, theta_warm, D[t])
   s = ig.abs().flatten(1).sum(1) # IG score of each filter
   L = select(s) # layer- then filter-level, Eq. (4)-(7)
   L = expand(L, theta_warm) # filter diversity expanding, Eq. (8)

   # continual filter learning (Sec. 3.3)
   B_prev = B
   B = cat([B_prev, theta_warm[L]])
   for x, y in loader(D[t]):
      g = mm(mm(B.T, W1), W2).T # generate filters from the bank
      if len(B_prev) >= len(g):
         g = g + 0.1 * attn(g, B_prev) # cross-task attention, Eq. (13)-(14)
      lam = sigmoid(w)
      theta[L] = (1 - lam) * theta_warm[L] + lam * g
      y_hat, e = f(theta, x)
      loss = l1(y_hat, y) + 0.1 * l_proto(e, P) + 0.01 * l_fft(y_hat, y) # Eq. (18)
      loss.backward()
      update(W1, W2, w)
      P[t] = 0.9 * P[t] + 0.1 * e.mean(0)

   B[-len(g):] = theta[L] # keep the learned filters for later tasks
   save(t, L, theta[L])
```

task is initialized by EMA; during the remaining 500 iterations, the prototypical contrastive loss is added, using the new prototype and all stored ones.

Continual filter learning. After localization, all weights are restored to θ^{t-1} except the localized filters, which keep their warmed-up values. For every selected convolution, these filters are appended to its filter bank $\mathcal{B}$, and a rank-8 generator (W_1, W_2) with a learnable mixing weight λ is created. The localized filters are then replaced by $(1-\lambda)\,\theta + \lambda\,\tilde{\theta}$, where $\tilde{\theta}$ is generated from the filter bank (Alg. 1). Only W_1, W_2 and λ are trained, for 150k iterations with the loss $10\,\ell_1 + \mathcal{L}_{\mathrm{FFT}} + 0.1\,\mathcal{L}_{\mathrm{proto}}$, AdamW, a constant learning rate and gradient clipping at 1.0, while the rest of the backbone stays frozen. After training, the generated filters are written into the backbone, so each task is deployed as a plain HINT model and only its localized filters need to be stored.

Cross-task attention. When generating filters for a convolution whose bank holds at least as many filters as are being generated, the generated filters attend to the bank: queries and keys are projected to $\min(32, d/2)$ dimensions, values keep the filter dimension d, and the attention output is added to the generated filters with a weight of 0.1. In our implementation, these projections keep their random initialization and are not updated during training. Since the generated filters are written into the backbone after each task, the attention brings no extra computation at inference.

Prototypical contrastive learning operates on features extracted from bottleneck layers. Specifically, we tap into the deepest encoder stage of the backbone, yielding a feature tensor of shape (B, C_b, H', W') where C_b=256 is the bottleneck channel dimension. A global average pooling layer then reduces this tensor to a compact (B, C_b) embedding, and both the EMA prototype update and the contrastive objective are computed. This design ensures that prototypes capture degradation-specific representations at a semantically rich abstraction level, rather than operating in the low-dimensional RGB output space where different restoration tasks would be largely indistinguishable, since all tasks converge toward similar clean-image statistics in the output domain.

Algorithm. Alg. 1 summarizes RwF in a PyTorch-like style. The first task uses the released model and defines the IG baseline. For each new task, the backbone is first warmed up on the new data,

Table 5: Task identification at inference on the four-task public benchmark. "PSNR change" is the PSNR obtained with the task predicted by the prototype router minus the PSNR obtained with the ground-truth task identity.

Metric	Deraining	Dehazing	Desnowing	LLE	Average
Routing accuracy (%)	98.0	90.4	100.0	100.0	97.1
PSNR change (dB)	-0.10	-1.18	0.00	0.00	-0.32

Table 6: Filter consumption and parameter overhead when the number of tasks grows to ten. All numbers are percentages.

Statistic	Number of learned tasks				
	2	4	6	8	10
Filters localized by the new task	2.05	2.44	2.87	2.95	2.67
Reused from previous tasks	0.0	54.1	71.9	81.1	88.3
Edited filters, random selection ↓	2.05	6.71	10.49	13.43	15.38
Edited filters, RwF (ours) ↓	2.05	5.25	6.83	7.77	**8.35**
Stored params, separate models ↓	100	300	500	700	900
Stored params, RwF (ours) ↓	2.66	10.02	17.29	25.40	**33.07**

and the key filters are localized in a coarse-to-fine manner and expanded with a few dissimilar filters. Then, all other weights are restored, and only a small generator per selected convolution is trained to produce the key filters from the filter bank. Finally, the generated filters are written into the backbone and stored together with their positions, and the model waits for the next task.

Task identification at inference. Since each task is served by its own localized filters, the model needs the task identity of a test image. Instead of requiring it as an input, we infer it with a lightweight prototype-based router. The degraded image, resized to 256×256, is passed through the encoder of the frozen first-task model, and its global-average-pooled latent feature (384-d) is used as the descriptor. For each task, a Gaussian with a shrinkage covariance is fitted to the descriptors of its training images and serves as the task prototype; a test image is assigned to the task with the highest likelihood, and the filters of that task are assembled into the backbone. The router requires no training and stores only a mean and a covariance per task. As shown in Tab. 5, routing is accurate on all restoration tasks.

D Additional Experimental Analysis

Capacity analysis. To test whether RwF remains effective as tasks accumulate, we extend the four-task benchmark of Sec. 4.3 to ten tasks on HINT by appending deblurring and denoising with $\sigma \in \{10, 15, 25, 35, 50\}$ (tasks 6–10 are trained for only 10k iterations); Tab. 6 summarizes the statistics. Each new task localizes 2.05%–2.95% of the backbone's filters regardless of its position in the sequence, and later tasks mostly reuse filters localized before (88.3% reuse at the tenth task), so the filters edited by all ten tasks cover only 8.35% of the backbone, against 15.38% for a random selection of the same size per layer; no layer is exhausted, the most occupied one (among layers with at least 32 filters) using 60.4% of its filters. Since each task stores its own values of the localized filters plus its adapter, the stored parameters grow linearly: after ten tasks RwF stores 33.1% of the backbone in total, i.e., 3.7% per task, comparable to a rank-8 LoRA (3.5%) and far below one model per task (900%). Meanwhile, all 22.0M convolution and linear weights outside the edited filters (88.6% of the backbone) stay unchanged from the first task to the tenth, so the performance of every learned task is preserved no matter how many tasks follow.

Effect of Prototypical Contrastive Learning To intuitively demonstrate the efficacy of the proposed prototypical contrastive loss $\mathcal{L}_{proto}$, we visualize the latent feature representations of different degradation tasks using t-SNE (Van der Maaten & Hinton, 2008), as shown in Fig. 8. Without $\mathcal{L}_{proto}$, the feature distributions of distinct tasks (e.g., deraining, dehazing, and desnowing) severely overlap, which inevitably leads to gradient interference and catastrophic forgetting during sequential training.

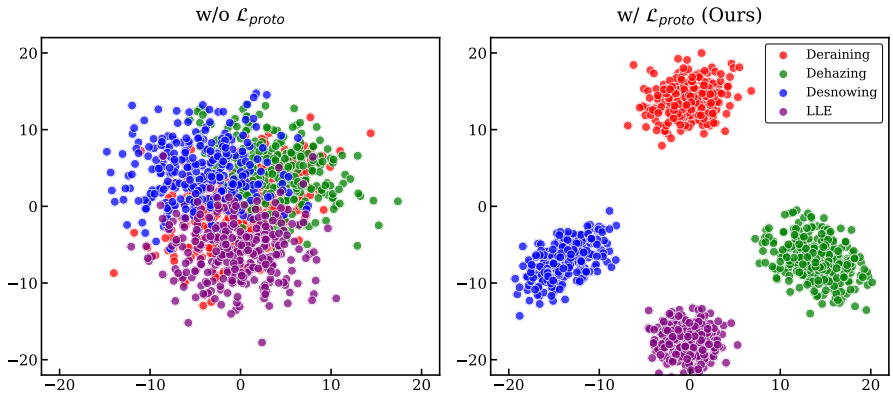

Figure 8: t-SNE visualization of latent feature distributions across four restoration tasks, comparing models trained with and without prototypical contrastive learning.

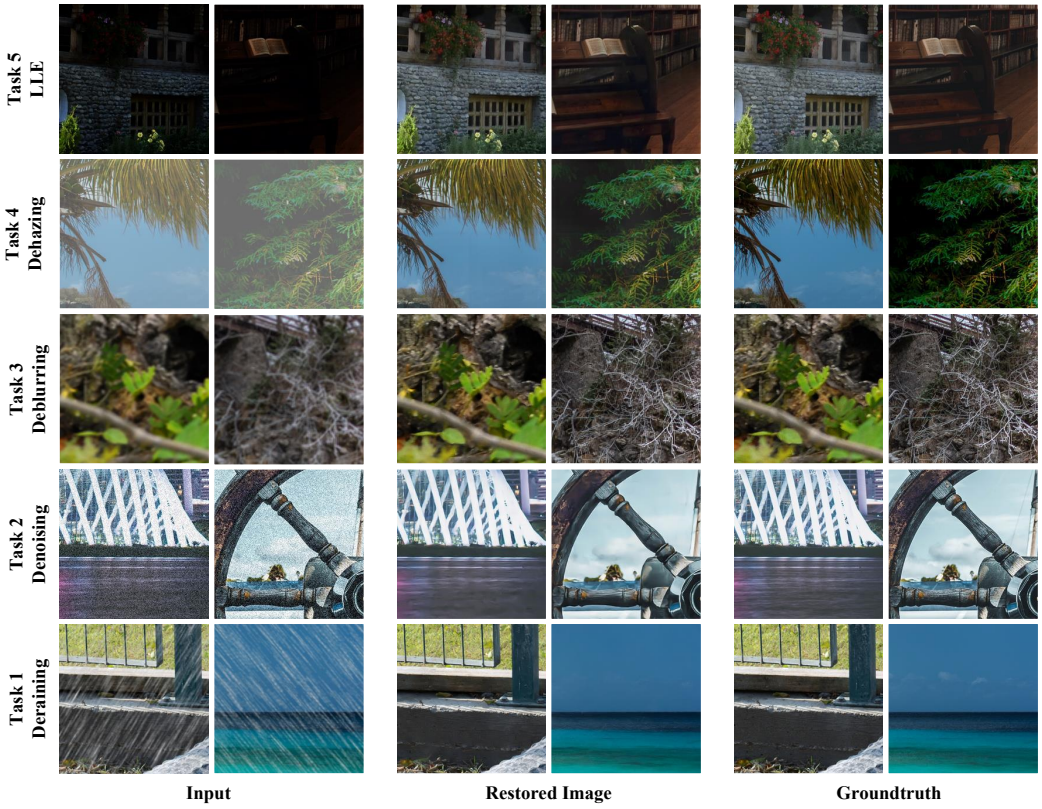

Figure 9: Visualization of restored images on randomly selected samples from test set after expanding to 5 restoration tasks in preliminary experiments, with the training sequence of deraining, denoising, deblurring, dehazing and LLE.

In contrast, by pulling features toward their respective task prototypes and pushing them away from historical ones, our RwF framework effectively enforces task-discriminative representations by prototypical contrastive learning. The resulting latent space exhibits well-separated clusters for each degradation pattern, explicitly mitigating negative transfer across sequential tasks.

Preliminary experiments with more tasks. To further validate the scalability and robustness of our proposed RwF framework over longer continual learning sequences, we extend the preliminary experiments from original three tasks to a five-task sequence. Specifically, we sequentially introduce

Table 7: Quantitative results after expanding to 5 restoration tasks, with the training sequence of deraining, denoising, deblurring, dehazing and LLE.

Method	Deraining		Denoising		Deblurring		Dehazing		LLE		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SFT	13.64	0.633	14.85	0.521	15.60	0.627	11.70	0.713	**28.75**	**0.939**	16.91	0.687
PiggybackGAN (Zhai et al., 2020)	32.51	0.895	29.54	0.824	31.46	0.878	27.80	0.867	27.45	0.882	29.75	0.869
RwF-MPRNet (Ours)	**38.52**	**0.942**	**33.69**	**0.874**	**34.49**	**0.913**	**32.01**	**0.920**	28.08	0.916	**33.36**	**0.913**

Table 8: LPIPS and DISTS (lower is better) on the four-task public benchmark with the order of deraining, dehazing, desnowing, and low-light enhancement (LLE), evaluated on all tasks after learning the last task. Best results within each backbone are in bold.

Method	Deraining		Dehazing		Desnowing		LLE		Average	
	LPIPS	DISTS	LPIPS	DISTS	LPIPS	DISTS	LPIPS	DISTS	LPIPS	DISTS
Degraded input	0.271	0.182	0.104	0.123	0.356	0.209	0.519	0.352	0.313	0.216
DiffUIR (zero-shot) (Zheng et al., 2024)	0.120	0.097	**0.008**	**0.014**	0.147	0.101	**0.139**	**0.125**	0.104	0.084
SFT-DiffUIR (Zheng et al., 2024)	0.272	0.183	0.076	0.078	0.270	0.170	0.165	0.139	0.196	0.143
RwF-DiffUIR	**0.033**	**0.038**	**0.008**	**0.014**	**0.145**	**0.100**	0.147	0.126	**0.083**	**0.069**
SFT-HINT (Zhou et al., 2025)	0.327	0.217	0.272	0.209	0.405	0.232	**0.142**	**0.124**	0.287	0.195
SFT + Rehearsal-HINT	0.034	0.046	0.018	0.025	0.148	**0.111**	0.156	0.136	0.089	**0.079**
RwF-HINT	**0.011**	**0.018**	**0.017**	**0.024**	**0.143**	0.113	0.172	0.159	**0.086**	**0.079**

dehazing and low-light enhancement following the initial deraining, denoising, and deblurring tasks. As demonstrated in Tab 7 and Fig 9, RwF maintains its high-fidelity restoration performance across all five tasks. This demonstrates its superior plasticity and stability even under extended and more complex continual restoration scenarios.

Perceptual quality. Tab. 8 reports LPIPS (Zhang et al., 2018) and DISTS (Ding et al., 2022) to compare perceptual quality between competing methods, where both are consistent with PSNR and SSIM. SFT severely degrades the perceptual quality of the earlier tasks, e.g., the LPIPS of deraining rises to 0.327 after the last task, which is even worse than the degraded input (0.271), whereas RwF-HINT keeps it at 0.011. RwF-HINT achieves the lowest average LPIPS among the HINT-based methods without storing any past data, while SFT with rehearsal needs to reuse training images per task to reach a similar level. On LLE, the last task, SFT reaches a lower LPIPS since it is fine-tuned without any constraint. For DiffUIR, RwF reduces the average LPIPS of the zero-shot model from 0.104 to 0.083.

More Visualizations of main experiments. Due to space constraints in the main manuscript, we provide more comprehensive qualitative comparisons on the four-task public restoration benchmark (Rain100L, SOTS, Snow100K, and LOLv2-Real) in Fig. 10 to Fig. 11. As consistently observed, sequential fine-tuning (SFT) and iCaRL often produce severe blending artifacts and color shifts when evaluated on previously learned tasks. Conversely, our RwF seamlessly recovers sharp structural edges, rich textures, and natural color distributions across all diverse degradations.

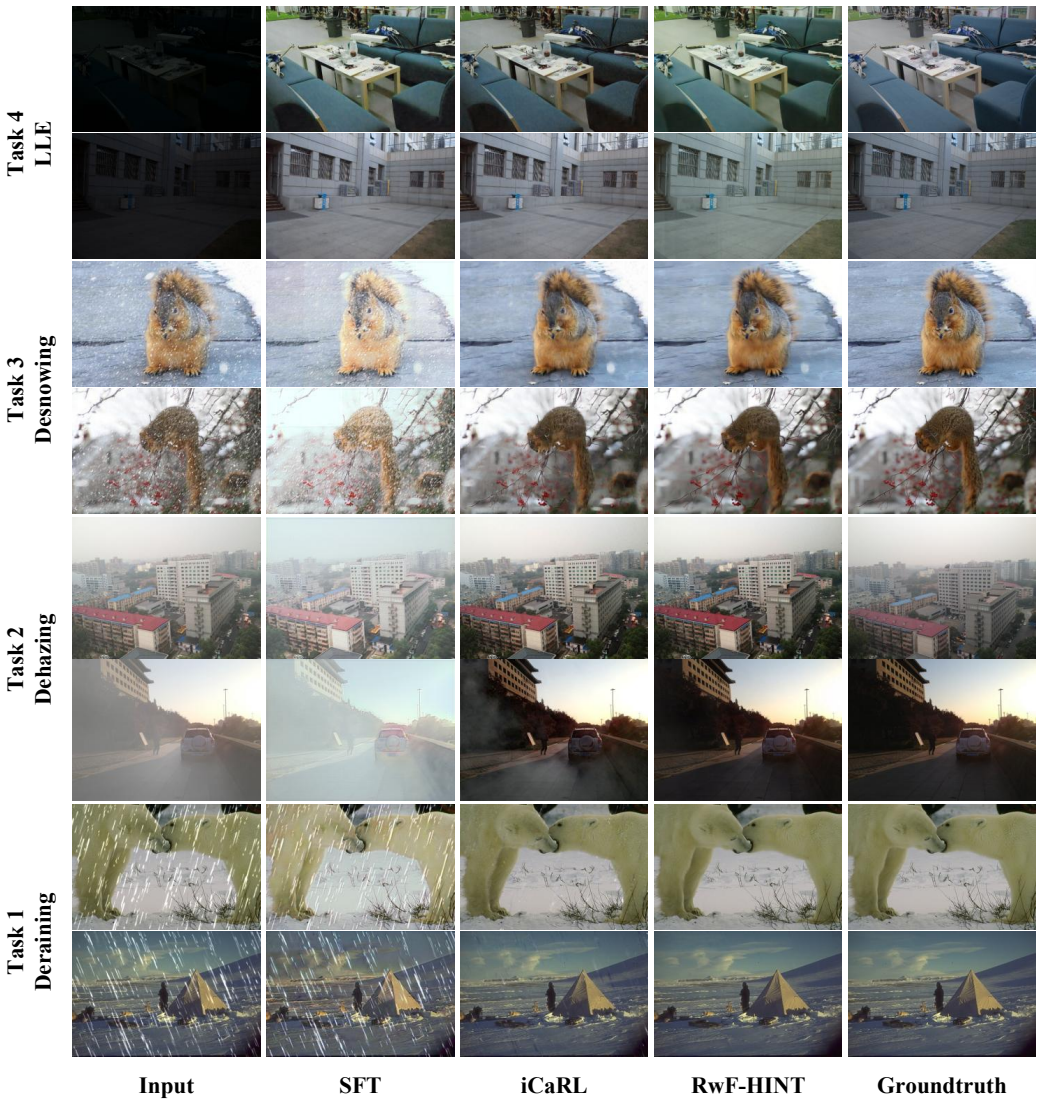

Figure 10: Visualization of restored images on randomly selected samples from test set of four-task public benchmarks.

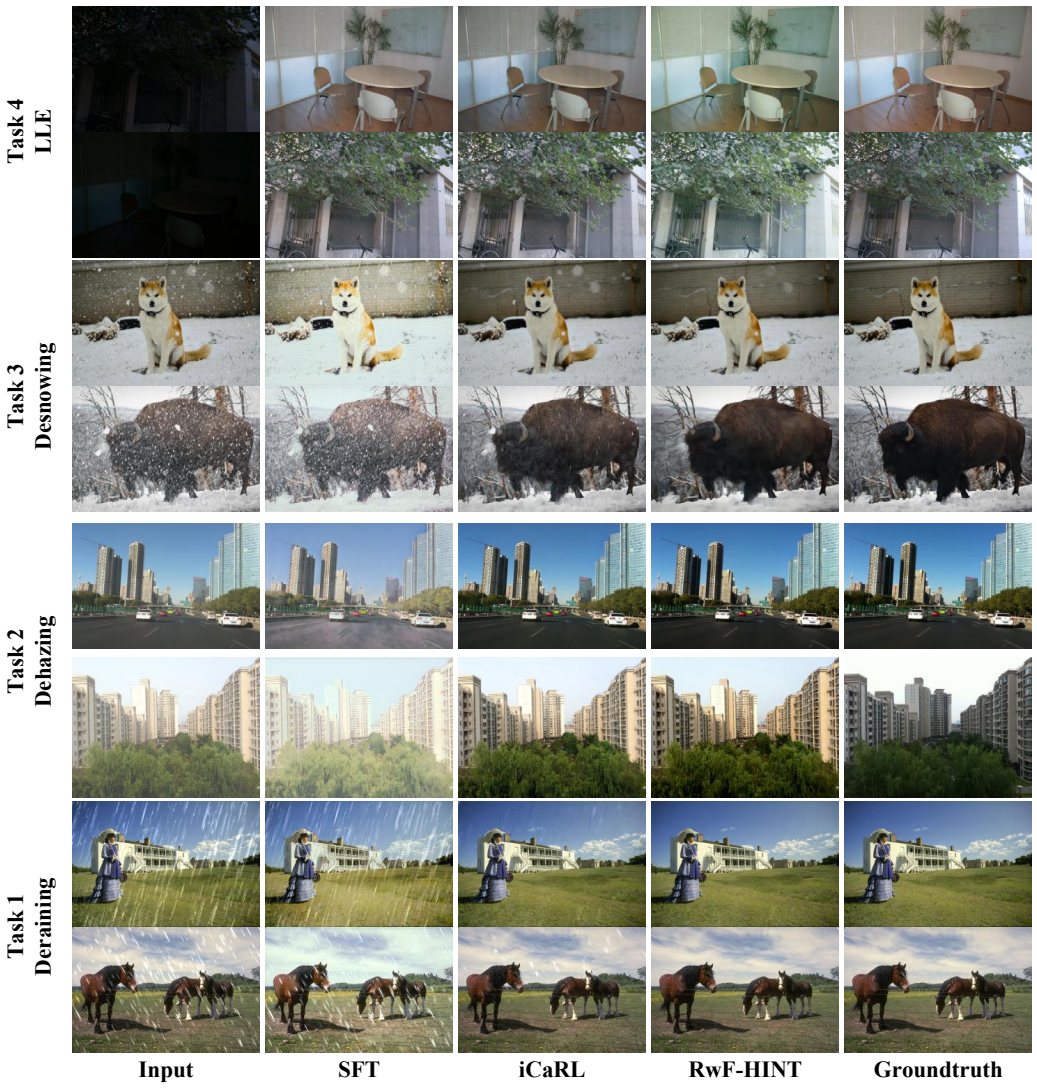

Figure 11: Visualization of restored images on randomly selected samples from test set of four-task public benchmarks.