A Minimalistic Unified Framework for Incremental Learning across Image Restoration Tasks

Xiaoxuan Gong
School of Artificial Intelligence and Automation, HUST
Huazhong University of Science and Technology
Wuhan, China
gongxiaoxuan286@gmail.com

Jie Ma*
School of Artificial Intelligence and Automation, HUST
Huazhong University of Science and Technology
Wuhan, China
majie@hust.edu.cn

Abstract

Existing research in low-level vision has shifted its focus from "one-by-one" task-specific methods to "all-in-one" multi-task unified architectures. However, current all-in-one image restoration approaches primarily aim to improve overall performance across a limited number of tasks. In contrast, how to incrementally add new image restoration capabilities on top of an existing model — that is, task-incremental learning — has been largely unexplored. To fill this research gap, we propose a minimalistic and universal paradigm for task-incremental learning called MINI. It addresses the problem of parameter interference across different tasks through a simple yet effective mechanism, enabling nearly forgetting-free task-incremental learning. Specifically, we design a special meta-convolution called MINIconv, which generates parameters solely through lightweight embeddings instead of complex convolutional networks or MLPs. This not only significantly reduces the number of parameters and computational overhead but also achieves complete parameter isolation across different tasks. Moreover, MINIconv can be seamlessly integrated as a plug-and-play replacement for any convolutional layer within existing backbone networks, endowing them with incremental learning capabilities and boosting their multi-task overall performance. Therefore, our method is highly generalizable. Finally, we demonstrate that our method achieves state-of-the-art performance compared to existing incremental learning approaches across five common image restoration tasks. Moreover, the near forgetting-free nature of our method makes it highly competitive even against all-in-one image restoration methods trained under joint learning. Our code is available at `https://github.com`.

1 Introduction

The core challenge of all-in-one(AIO) image restoration lies in the need to accomplish diverse feature extraction and image manipulation tasks using a single set of fixed parameters, which inevitably leads to parameter conflicts between different tasks. To alleviate this issue, most existing methods introduce additional information (prompts) to guide the model in handling different types of degradations (as Figure 1a), such as [1, 2, 3, 4, 5]. Alternatively, a more recent trend is to leverage the priors from large-

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

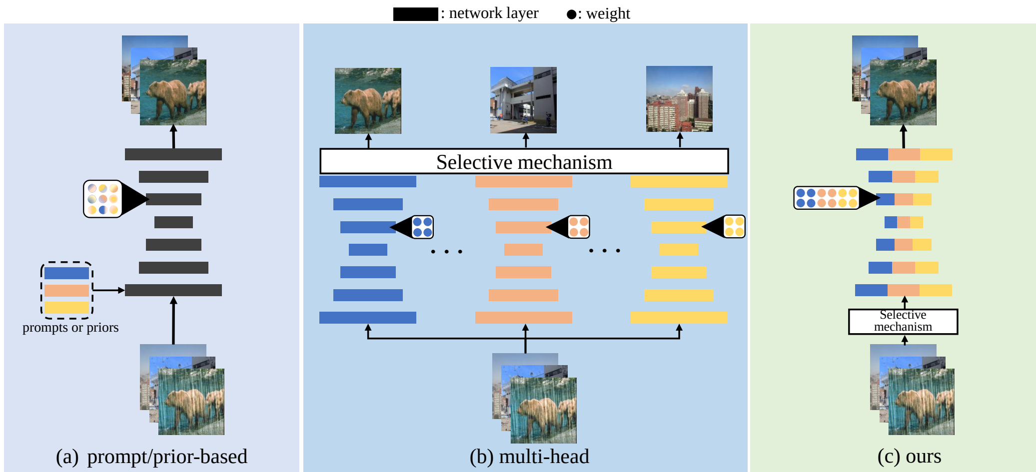

Figure 1: Illustration of different multi-task image restoration paradigms. Colors indicate distinct image restoration tasks such as deraining, dehazing, and raindrop removal etc. (a) prompt/prior-based methods: although additional prompt information is provided, the parameters are not isolated—shared weights are still responsible for multiple tasks; (b) muiti-head methods: parameter isolation is achieved by parallelizing multiple task-specific subnetworks, but the overall structure becomes bloated and difficult to deploy; (c) MINI: parameter isolation is implemented at each layer within a single network, resulting in a lightweight architecture that is easier to deploy and transfer.

scale models as guidance, such as [6, 7, 8, 9, 10]. Although these methods have achieved impressive results, they do not fundamentally address the issue of parameter conflict across different tasks. As a result, they often suffer from significant performance degradation compared to task-specific models. Moreover, when performing incremental learning over new tasks, these methods are prone to severe catastrophic forgetting, leading to unacceptable overall performance. This demonstrates that parameter conflict is a shared bottleneck for both all-in-one image restoration and task-incremental learning.

To address the parameter conflict issue, a natural idea is to train multiple models, each responsible for a different task, and then select different pathways through some selection mechanism, thereby achieving parameter isolation across tasks (as Figure 1b). However, such a multi-head approach means that a complete new pathway must be added for each new task, resulting in increased parameters and computational overhead. In addition, some regularization-based approaches have been proposed in incremental learning to alleviate catastrophic forgetting[11, 12, 13]. However, these methods are often limited in effectively addressing parameter conflicts. A more elegant solution is to employ dynamic parameters within a single model instead of fixed ones, allowing the parameters to adapt based on the model input. Such approaches are known as dynamic convolution or meta convolution [14, 15, 16, 17]. These methods aim to use a small network, referred to as a meta-network, to generate the convolutional weights of the main network, thereby avoiding the parameter conflicts caused by fixed weights. Nevertheless, these meta-networks introduce significant computational overhead and are notoriously difficult to optimize during training. Furthermore, paradoxically, since the meta-networks themselves rely on fixed parameters, they are also prone to parameter conflicts when handling diverse inputs.

For addressing the above issues, we propose the ***M****inimalistic* ***I****ncremental* ***N****etwork for* ***I****mage Restoration*(**MINI**), a novel and lightweight universal architecture. MINI's core component is a specialized Meta Convolution module, which we call **MINIconv**. Unlike vanilla meta convolutions, it does not introduce any additional computational overhead. Instead, it achieves complete parameter isolation between different tasks through a selective embedding mechanism, thus possessing an extremely simple structure. Moreover, this embedding-based structure is naturally well-suited for task-incremental learning, and when combined with a simple query mechanism, it can achieve near-forgetting-free task expansion. Finally, we design a specialized embedding regularization method to enhance the robustness of MINI. Extensive comparative experiments demonstrate that our method surpasses all existing incremental learning approaches in task-incremental settings, achieving state-

Table 1: Qualitative comparison among different types of multi-task image restoration methods

Methods/Property	parameter-isolated	low-overhead	incrementally-adaptive	optimization-friendly
Prompt/Prior-based	×	✓	×	✓
Multi-head	✓	×	✓	✓
Meta-Conv	✓	×	×	×
MINI (Ours)	✓	✓	✓	✓

of-the-art performance. Remarkably, under our MINI architecture, the overall multi-task performance of most existing image restoration methods is significantly improved. Overall, the main contributions of this paper are as follows:

- To the best of our knowledge, MINI is one of the first attempts to explore task-incremental learning in the field of image restoration. It serves as a strong and minimal baseline that can inspire future research in this emerging sub-task.
- We propose MINI with a minimalistic design that achieves almost complete parameter isolation across different tasks. Moreover, the proposed MINIconv can be seamlessly integrated into any existing image restoration backbone, endowing it with task-incremental learning capability and boosting their multi-task overall performance, while introducing negligible additional computational cost.
- To further enhance the robustness of MINI, we introduce a task-aware embedding consistency regularization tailored to its structure.
- MINI achieves state-of-the-art performance compared to existing task-incremental learning methods, and is even competitive with AIO image restoration approaches trained under joint learning.

2 Preliminary: Meta Convolution in Image Restoration

Meta convolution (MetaConv) is a class of parameter-adaptive techniques that dynamically generate convolutional weights based on task-specific or input-conditioned information. First introduced in the context of dynamic filter networks[18], and further popularized by approaches such as HyperNetworks[17] and Dynamic Convolution [19], MetaConv has recently been applied to a range of low-level vision tasks, such as super-resolution[20]. The core idea is to replace fixed convolutional weights with weights produced by a small auxiliary network—known as a meta-network—which allows the model to adapt to varying tasks or degradations by conditioning on additional embeddings or features.

While conceptually appealing, existing MetaConv methods face several significant limitations:

- Increased computational complexity: The meta-network itself often comprises multi-layer perceptrons or lightweight convolutional sub-networks. These introduce considerable overhead during both training and inference[17].
- Optimization difficulties: MetaConv introduces a nested dependency between the generated weights and the meta-network's parameters, which complicates gradient flow and frequently results in unstable or slow convergence[21].
- Static meta-parameters: Paradoxically, while MetaConv aims to mitigate task interference by generating dynamic weights, the meta-network itself is typically fixed once trained. This means the meta-network may still suffer from parameter conflicts when facing multiple tasks or distribution shifts.

These issues limit the scalability and robustness of MetaConv, particularly in the context of incremental image restoration, where tasks arrive sequentially and require both parameter isolation and computational efficiency. Addressing these challenges is the motivation behind our proposed architecture. Table 1 presents a qualitative comparison of various types of multi-task image restoration methods.

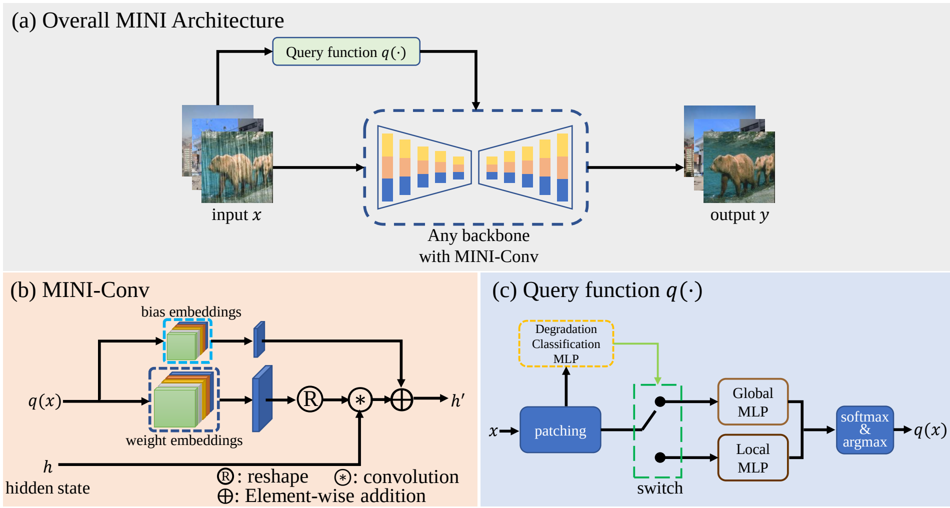

Figure 2: An overview of the MINI architecture. MINI can be built upon any existing image restoration backbone by simply replacing the original convolutional layers with our proposed MINIconv and introducing a lightweight degradation query mechanism. Subfigure (b) shows the structure of MINIconv, which consists of only two parallel embedding pools without any complex components, as detailed in Section 3.1. Subfigure (c) illustrates our lightweight degradation query module, which provides specific task IDs for MINIconv to select the corresponding embeddings for use, as detailed in Section 3.2.

3 Minimalistic multi-task image restoration architecture

Our proposed MINI architecture is illustrated in Figure 2(a). As shown, the design is remarkably simple and intuitive, and can be built upon any existing image restoration backbone. The only modifications required are to replace the standard convolutions in the original backbone with MINIconvs and to introduce a lightweight query function. In other words, the MINI framework is essentially "MINIconv + query + any backbone." It is worth emphasizing that our approach does not focus on designing complex novel structures or sub-modules. In contrast, we aim to address the problem of parameter conflicts across multiple tasks using the most minimalistic paradigm possible.

3.1 MINIconv: embeddings are all your need

Classical regularization-based methods[11, 12, 13, 22] in incremental learning focus on constraining the model to update parameters that are more relevant to the current task, while minimizing changes to those deemed less important. We consider this essentially a form of "soft" parameter isolation. Inspired by these methods, we seek a **hard parameter isolation mechanism**, where only a subset of parameters is deterministically updated during incremental learning, while the remaining parameters are completely excluded from training. In this way, when training the model on new tasks, the weights associated with previously learned tasks remain completely unaffected.

Unfortunately, it is difficult to implement the aforementioned hard isolation mechanism in a standard convolutional layer, as convolution kernel is overly compact and the convolution operation itself is inherently continuous and sliding. To address this, we propose MINIconv, a specialized and flexible convolutional structure, as illustrated in Figure 2(b). The core components of MINIconv only consist of two embedding pools, each composed of a fixed number of embeddings with desirable separability. In the PyTorch framework, they can be conveniently implemented using *nn.Embedding*. Each weight embedding pool is set to have a size of $(C_{in} \times C_{out} \times K^2)/G$, where C_{in} is the number of input channels of the hidden state h, C_{out} is the number of output channels, K is the kernel size, and G is the number of groups in the grouped convolution. And each bias embedding pool is set to have a size of C_{out}. Each pool consists of T embeddings of the same size, representing the maximum number of tasks the model can accommodate. During training and inference, one embedding is selected from each of the two pools via the query function (detail in Section 3.2). Then after a simple reshaping, these two embeddings are used to perform standard convolution operations on the hidden state h, and the remaining embeddings are excluded from both forward and backward propagation, thereby achieving complete parameter isolation. The complete computation process of MINIconv can be formulated as follows:

$$\begin{cases} t = q(x), \\ W_t = reshape(e_w^{(t)}) \in \mathbb{R}^{C_{out} \times C_{in} \times K \times K}, \\ b_t = e_b^{(t)} \in \mathbb{R}^{C_{out}}, \\ h' = W_t \circledast h + b_t, \end{cases} \tag{1}$$

where "$\circledast$" denotes convolution operation, $e_w^{(t)}, e_b^{(t)}$ denote t-th embedding of weight embedding pool and bias embedding pool respectively, $q(\cdot)$ is the query function.

Intuitively, a specific task (e.g., deraining or dehazing) exclusively uses its assigned weight and bias embeddings. When a new task arrives, it only needs to be allocated an unused embedding pair as its dedicated parameters. Therefore, this mechanism is highly suitable for task-incremental learning. Interestingly and perhaps surprisingly, although MINIconv increases the total number of convolutional parameters by a factor of T, its minimalist select-and-use mechanism incurs almost no additional computational cost. This is because only the parameters corresponding to a single standard convolution kernel are actually involved during each forward pass. Therefore, MINIconv can be extensively applied throughout the network, unlike traditional MetaConv approaches that rely on auxiliary networks to generate dynamic kernels, which can lead to considerable computational overhead when used widely.

3.2 Degradation query mechanism

As described in the previous section, MINIconv requires a query mechanism to determine the task type of the input image and select the corresponding embedding accordingly — essentially serving as a degradation classifier. Some promising related works already be proposed, such as DA-CLIP[6]. However, to adhere to the principle of minimalism, we propose a lightweight degradation classification module, as illustrated in Figure 2(c). We observe that most image degradations can be roughly categorized into global (e.g., low-light, blur) and local (e.g., rain, raindrops, fog). Therefore, we first divide the input image into patches and feed them into a global/local classification MLP, which outputs a two-dimensional vector. The degradation type is then determined via an argmax operation. Based on this result, the patched image is further routed to either the global or local MLP branch for more fine-grained degradation classification. The final output vector is passed through a softmax function, and the task ID is obtained via an argmax operation. The entire process can be described as follows:

$$\begin{cases} \{x_i\}_{i=1}^{N} = \text{Patching}(x), \\ z = MLP_{global-local}(\text{Aggregate}(\{x_i\})) \ \in \mathbb{R}^2, \\ t_{mode} = \text{argmax}(\sigma(z)) \ \in \{1,2\}, \\ v = \begin{cases} MLP_{global}(\{x_i\}), \ if \ t_{mode} = 1 \\ MLP_{local}(\{x_i\}), \ if \ t_{mode} = 2 \end{cases} \ \in \mathbb{R}^T \\ t_{id} = \text{argmax}(\sigma(v)), \end{cases} \tag{2}$$

where N is the number of image patches, $\sigma(\cdot)$ denotes the softmax operation, and t_{id} denotes the final task ID obtained. Before training the main network, the entire query module can be simply pretrained using the following loss function:

$$\mathcal{L}_{query} = -\text{log}\left(\frac{\text{e}^{v_m}}{\sum_{j=1}^{T}\text{e}^{v_j}}\right) - \lambda_{query}\text{log}\left(\frac{\text{e}^{z_n}}{\sum_{i=1}^{2}\text{e}^{z_i}}\right), \tag{3}$$

where $n \in \{1,2\}$ is the ground-truth label indicating whether the degradation is global or local, $m \in \{1,2,...,T\}$ is the ground-truth task ID, and λ_{query} is a balance coefficient. In our early experiments, we attempted to build the degradation classifier using either a single MLP or a series of MLPs in sequence. However, we found that the classifier consistently struggled to distinguish certain types of degradations, such as blur and fog. To address this issue, we proposed a two-stage degradation classification mechanism — performing coarse classification first, followed by fine-grained classification — which led to the design of our current query module. This simple yet effective change improved the classification accuracy by approximately 15

3.3 Embedding-consistency regularization

Although the above MINIconv and query mechanism are sufficient to construct a complete multi-task image restoration framework, we observe in practice that the overall performance of MINI is highly sensitive to the accuracy of the query mechanism. Once the degradation type of a sample is misclassified — even for a small number of samples — it can significantly degrade the overall performance. Therefore, in training phase, to enhance the fault tolerance and robustness of MINI, we introduce a specialized embedding-consistency regularization (ECR) method, as formulated below:

$$\begin{cases} \bar{W}^{(l)} = \frac{1}{t_{now}-1} \sum_{t=1}^{t_{now}-1} W_t^{(l)}, \\ \bar{b}^{(l)} = \frac{1}{t_{now}-1} \sum_{t=1}^{t_{now}-1} b_t^{(l)}, \\ \mathcal{L}_{ecr} = \sum_{l=1}^{L} \|W_{t_{now}}^{(l)} - \bar{W}^{(l)}\|_2^2 + \sum_{l=1}^{L} \|b_{t_{now}}^{(l)} - \bar{b}^{(l)}\|_2^2 \end{cases} \tag{4}$$

where L is the total number of MINIconv layers in the model, $W_t^{(l)}, b_t^{(l)}$ denote the t-th weight embedding and bias embedding in the l-th MINIconv layer respectively, and t_{now} denotes the ID of the newly introduced task currently being trained, and $\bar{W}^{(l)}, \bar{b}^{(l)}$ denote the mean of the first $t_{now} - 1$ weight embeddings and bias embeddings in l-th MINIconv layer, respectively.

Intuitively, we expect the embedding of the new task to not differ significantly from those of existing tasks, encouraging the embeddings responsible for different tasks to remain as consistent as possible. In this way, even if the query mechanism makes an incorrect degradation prediction and selects the embedding of another task, the overall performance will not be significantly degraded. Overall, the total training loss of MINI is as follows:

$$\mathcal{L}_{all} = \mathcal{L}_{main} + \lambda_{ecr}\mathcal{L}_{ecr}, \tag{5}$$

where λ_{ecr} is the ECR regularization coefficient, $\mathcal{L}_{main}$ refers to common reconstruction losses such as L1, L2, or perceptual loss etc.

4 Experiments

4.1 Comparative experiment

Table 2: Quantitative comparison of several image restoration baselines trained on five datasets using one-for-one, joint learning, and incremental learning (MINI) strategies. The incremental learning is conducted in a sequential manner following the task order of rain → haze → blur → raindrop → low light. In all-in-one manner, the symbol * indicates that the data is reported from the original paper. "↑" indicates that a higher value is better for the metric, while "↓" indicates that a lower value is preferable. In the MINI framework, the metric values that show improvement compared to the all-in-one setting are highlighted in **bold**.

datasets		R100H (rain)			RESIDE-6k (haze)			GoPro (blur)			Raindrop (raindrop)			LOLv2 (low light)		
methods / metrics		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
one-for-one (task-specific)	MAXIM[23]	30.81	0.901	-	29.12	0.932	-	33.86	0.961	-	-	-	-	23.43	0.863	0.111
	NAFNet[24]	-	-	-	-	-	-	32.85	0.960	-	-	-	-	-	-	-
	Restormer[25]	31.46	0.904	-	-	-	-	32.92	0.961	-	-	-	-	-	-	-
	IR-SDE[26]	31.65	0.904	0.047	-	-	-	30.70	0.901	0.064	-	-	-	-	-	-
	DA-CLIP[6]	33.91	0.926	0.031	30.16	0.936	0.030	30.88	0.903	0.058	31.50	0.944	0.056	23.77	0.830	0.083
all-in-one (joint learning)	AirNet[1]	30.21	0.905	0.145	27.94	0.912	0.041	27.85*	0.892*	-	27.13	0.890	0.089	21.05	0.862	0.124
	PromptIR[2]	31.02*	0.914*	-	29.57	0.923	0.045	28.05	0.901	0.068	28.36	0.912	0.074	21.96	0.886	0.118
	MAXIM[23]	29.34	0.886	0.075	29.15	0.914	0.039	29.51	0.905	0.063	27.90	0.895	0.081	21.35	0.875	0.121
	NAFNet[24]	30.42	0.875	0.066	27.09	0.941	0.037	28.03	0.856	0.074	29.75	0.916	0.051	20.97	0.871	0.105
	Restormer[25]	30.59	0.893	0.086	28.12	0.957	0.041	29.32	0.879	0.063	29.87	0.918	0.042	21.37	0.873	0.111
	IR-SDE[26]	30.95	0.892	0.067	29.33	0.950	0.038	28.85	0.881	0.068	30.34	0.926	0.032	21.94	0.882	0.109
	DA-CLIP[6]	31.51	0.923	0.052	29.58	0.956	0.036	29.29	0.902	0.070	30.44	0.880	0.078	22.15	0.887	0.101
MINI (incremental learning)	AirNet[1]	**31.20**	**0.914**	**0.075**	**29.75**	**0.948**	**0.033**	**30.26**	**0.905**	0.063	**29.81**	**0.904**	**0.056**	**21.91**	**0.882**	**0.108**
	PromptIR[2]	**31.51**	0.912	0.064	**30.64**	**0.952**	**0.037**	**28.90**	**0.916**	**0.065**	**30.67**	**0.918**	**0.059**	21.94	0.885	**0.101**
	MAXIM[23]	**31.32**	**0.903**	**0.059**	**30.55**	**0.941**	**0.037**	**32.33**	**0.957**	**0.060**	**31.94**	**0.927**	**0.043**	**23.01**	**0.896**	**0.094**
	NAFNet[24]	**30.90**	**0.918**	**0.042**	**29.83**	**0.960**	**0.028**	**29.96**	**0.893**	0.095	**30.48**	0.912	**0.046**	**22.60**	**0.873**	0.108
	Restormer[25]	**31.39**	**0.901**	**0.040**	**30.15**	**0.968**	**0.024**	**32.09**	**0.924**	**0.056**	**31.54**	**0.923**	**0.039**	**22.56**	**0.884**	**0.105**
	IR-SDE[26]	**31.33**	**0.905**	**0.056**	**30.20**	**0.957**	**0.029**	**29.96**	**0.909**	**0.059**	**32.09**	**0.930**	0.035	**22.45**	**0.891**	**0.098**
	DA-CLIP[6]	**31.89**	**0.927**	**0.039**	**30.18**	0.949	**0.032**	**30.25**	**0.914**	**0.053**	**31.01**	**0.921**	**0.041**	**23.15**	**0.890**	**0.095**

4.1.1 Experiment setup

To demonstrate the effectiveness of our proposed MINI architecture for task-incremental image restoration, we conduct detailed comparisons on five datasets (five tasks) based on several existing image restoration methods, the five datasets are R100H[27], RESIDE-6K[28], GoPro[29], Raindrop[30], and LOLv2[31]. We evaluate image restoration performance using the following metrics: PSNR, SSIM[32], LPIPS[33]. We compare the overall performance of these methods under three training paradigms: one-for-one task-specific training, all-in-one joint training, and incremental training using the proposed MINI framework. For the all-in-one training setting, we train for 2000 epochs on a mixed dataset composed of the five aforementioned training sets. The learning rate follows a cosine annealing schedule with warm-up, peaking at 0.0002. The loss function consists of an L2 loss combined with a VGG16-based perceptual loss. It is important to **NOTE** that during actual training, the embeddings need to be manually initialized using He initialization[34]; otherwise, the model may struggle to converge. The training is conducted on two NVIDIA 2080ti GPUs.And due to the imbalance in the sizes of the five datasets, we adopt a common resampling strategy during training to ensure that each training batch contains a balanced number of images from each dataset. Specifically, since IR-SDE[26] and DA-CLIP[6] are diffusion-based methods that require more training iterations, they are trained for 3000 epochs. For MINI, we adopt an incremental learning strategy following the task sequence: R100H $\rightarrow$ RESIDE-6K $\rightarrow$ GoPro $\rightarrow$ Raindrop $\rightarrow$ LOLv2. For each baseline, we train on each dataset for 400 epochs under the same settings, while IR-SDE and DA-CLIP are trained for 600 epochs. During the training phase, the degradation query module and the main network are trained separately. The MINIconv layers in the main network take the ground-truth task ID as input instead of $q(x)$. The hyperparameter settings are as follows: $\lambda_{query} = 1$, $\lambda_{ecr} = 0.001$, $T = 5$. During inference, $q(x)$ is used as the input to MINIconv.

Table 3: Final performance comparison of model-agnostic generic incremental learning methods under NAFNet[24] backbone. L2P[35] and DualPrompt[36] use the backbone networks from their original papers, and their results are provided as reference for comparison.The training task sequence is: rain $\rightarrow$ haze $\rightarrow$ blur $\rightarrow$ raindrop $\rightarrow$ low light. The best results are highlighted in **bold**.

datasets	R100H (rain)			RESIDE-6k (haze)			GoPro (blur)			Raindrop (raindrop)			LOLv2 (low light)		
methods / metrics	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
LwF[13]	14.23	0.572	0.674	19.23	0.633	0.286	24.03	0.712	0.166	26.94	0.804	0.081	21.78	0.885	0.101
EWC[11]	13.36	0.512	0.584	19.51	0.653	0.271	21.36	0.623	0.255	23.45	0.740	0.156	22.12	0.871	0.111
SI[12]	11.36	0.496	0.612	18.24	0.597	0.365	22.34	0.603	0.311	24.97	0.769	0.163	21.24	0.869	0.113
MAS[22]	14.37	0.654	0.509	20.77	0.733	0.205	24.81	0.753	0.154	28.48	0.883	0.067	**22.73**	0.890	0.097
L2P[35]	17.06	0.694	0.201	22.73	0.763	0.137	24.53	0.788	0.105	24.61	0.908	0.052	21.39	**0.907**	0.090
DualPrompt[36]	16.03	0.682	0.124	20.03	0.733	0.136	24.48	0.792	0.116	27.04	0.874	0.071	21.90	0.904	**0.073**
MINI	**30.90**	**0.918**	**0.042**	**29.83**	**0.960**	**0.028**	**29.96**	**0.893**	**0.095**	**30.48**	**0.912**	**0.046**	22.60	0.873	0.108

4.1.2 Analysis

As shown in Table 2, under our MINI framework, the overall performance of various baseline methods on multi-task image restoration has been significantly improved. In some tasks, the performance even rivals that of their corresponding task-specific training versions. More importantly, MINI endows these methods with excellent incremental learning capabilities. Meanwhile, we take NAFNet[24] as baseline methods and compare MINI with existing model-agnostic generic incremental learning approaches, as shown in Table 3. The results show that our MINI design, based on "hard parameter isolation," effectively eliminates catastrophic forgetting and achieves state-of-the-art performance in task-incremental learning. In contrast, other methods suffer increasingly from catastrophic forgetting as the task sequence grows longer, ultimately leading to poor overall performance. The visual comparison is shown in Figure 3.

In addition, we compared meta-conv with our proposed MINI-conv under the same training settings in FLOPs and Rarams, and the results are shown in Table 4. We adopted the same NAFNet as the backbone, replaced its original standard convolutions, and conducted tests on images with a resolution of 256×256. It can be observed that, compared with standard 2D convolutions, meta-conv increases both FLOPs and Params, with a particularly large increase in Params, which is impractical for real-world applications. This limits the scalability of MetaConv within backbone networks. In contrast, our MINIconv has no impact on FLOPs, and its Params increase only linearly with the number of tasks, allowing it to replace standard 2D convolutions in the backbone on a large scale.

Table 4: Comparison between MetaConv and MINIconv. MetaConv generates convolution parameters using a single-layer MLP. The backbone network is NAFNet.

methods	FLOPs↓	Params↓
standard 2D-conv	22.56G	19.30M
MetaConv	30.22G	1814.1M
MINIconv	22.56G	96.5M

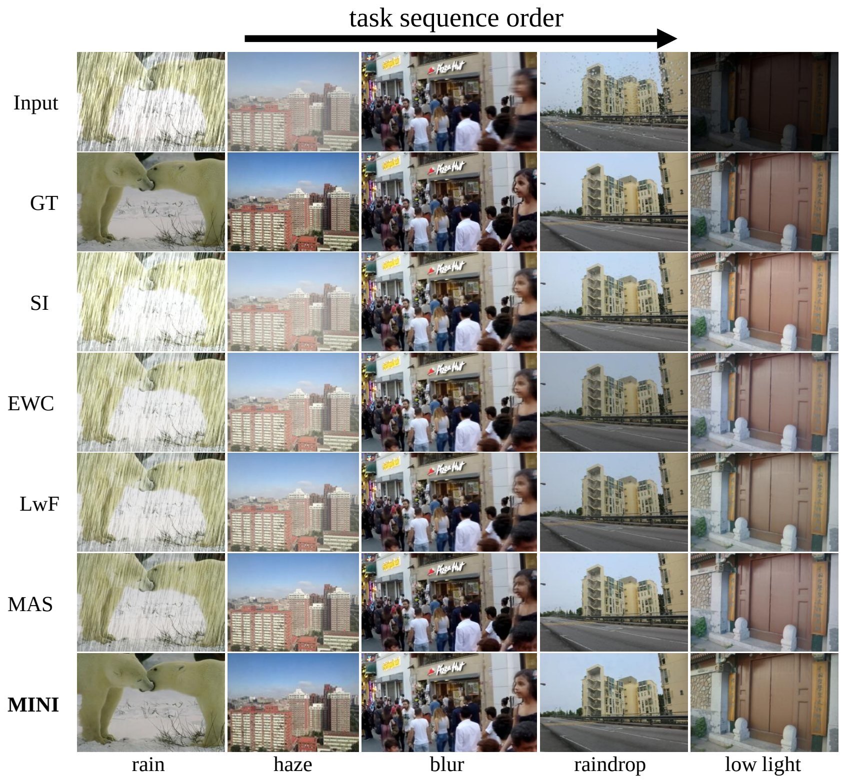

Figure 3: Visual comparison of model-agnostic generic incremental learning methods. Zoom in for details.

4.2 Ablation study

Embedding-consistency regulation. As described in Section 3.3, the accuracy of the query mechanism significantly affects the overall performance of MINI. To enhance the robustness and fault tolerance of MINI, we introduce an Embedding-Consistency Regularization (ECR). To validate its effectiveness, we intentionally inject a certain proportion of random incorrect degradation classifications into the model and compare its overall performance with and without the proposed regularization. Similarly, we train the MINI-based NAFNet on the five aforementioned datasets. The results are shown in Figure 4. Under different degradation classification error rates, ECR consistently leads to better overall performance compared to the case without ECR. Visual results under several misclassification cases are shown in Figure 5. It can be observed that even when degradation is misclassified, the model trained with ECR still maintains a certain level of image restoration capability, whereas the model trained without ECR exhibits almost no fault tolerance.

Our pretrained query module achieves an error rate of approximately 6%. While it is possible to adopt more advanced image classification models—such as the powerful DA-CLIP[6]—to further reduce the error rate, this would come at the cost of increased structural complexity and computational overhead. Therefore, applying ECR on top of a lightweight degradation classifier can be viewed as a better trade-off between performance and efficiency.

Task sequence order. To explore the impact of task training order on MINI, we train the MINI-based NAFNet under four different task sequence orders and compare the final performance, as shown in Table 5. The results show that, thanks to MINI's strong parameter isolation capability, changing the task sequence order has little impact on its final performance. The variation in average PSNR is within 0.3 dB, SSIM within 0.01, and LPIPS within 0.002. This demonstrates the robustness of the MINI architecture to task order.

It is worth noting that, a model with complete parameter isolation should, in theory, achieve consistent performance regardless of the training order. Although our MINI achieves near-complete parameter

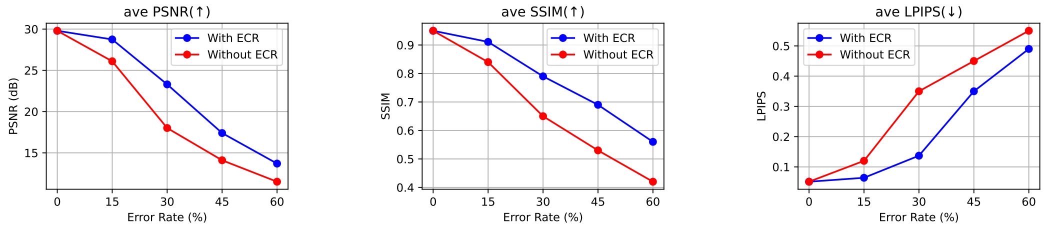

Figure 4: Overall performance comparison with and without ECR under different degradation classification error rates. The baseline architecture is the MINI-based NAFNet, with λ_{ecr} set to 0.001.

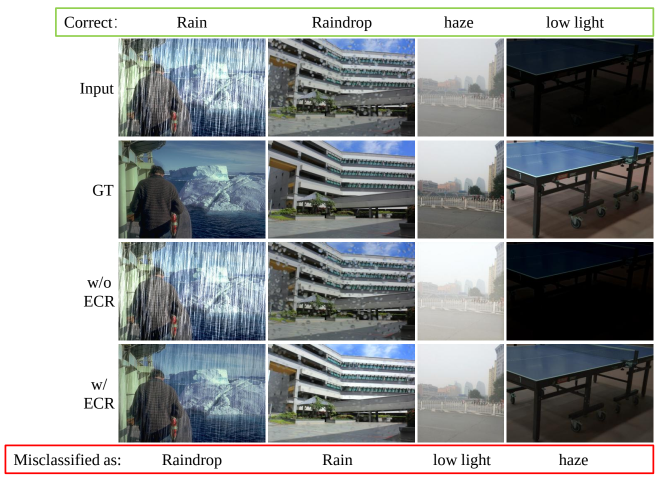

Figure 5: Visual comparison with and without ECR under incorrect degradation classification, based on the MINI-based NAFNet.

isolation, as discussed in Section 5 (Discussion), most baseline architectures still contain a small number of shared parameter components beyond standard convolutions—such as LayerNorm and nn.Parameter elements—which may introduce minor parameter conflict. This residual overlap is the primary reason behind the slight performance differences observed under different training orders. However, we found in practice that these shared parameters rarely lead to catastrophic forgetting. Therefore, in favor of architectural simplicity, we did not propose replacements like "MINI-LayerNorm" to fully isolate these components.

Table 5: Final overall performance of the MINI-based NAFNet under different training orders. The task IDs and corresponding datasets are as follows: 1: derain (R100H); 2: dehaze (RESIDE-6K); 3: deblur (RESIDE-6K); 4: raindrop removal (Raindrop); 5: low-light enhancement (LOLv2).

orders/ metrics	ave PSNR↑	ave SSIM↑	ave LPIPS↓
1→2→3→4→5	28.754	0.911	0.0632
5→4→3→2→1	28.612	0.912	0.0614
1→3→2→5→4	28.518	0.908	0.0619
3→2→4→1→5	28.625	0.914	0.0628

5 Discussion

Other parameter components except for convolutional layers.

In most CNN-based backbones, in addition to convolutional layers, there are also some smaller parameter components such as *LayerNorm* and *nn.Parameter* etc. Although these components can also suffer from parameter conflict across different tasks, we find in practice that converting them into "embedding-based" forms—similar to MINIconv—does not lead to significant improvements in multi-task performance (detailed in Appendix B). This is mainly because the parameter norms of these components do not vary substantially across tasks, meaning that even without explicit parameter isolation, they have limited impact on the overall performance. The application of the MINI architecture to transformer-based backbones will be explored in our future work. In addition, a preliminary test result on SwinIR[37] can be found in Appendix C.

Limitations of MINI.

Despite the strong performance and efficiency demonstrated by the proposed MINI architecture in task-incremental image restoration, there remain several limitations worth noting.

First, the number of tasks that MINI can support is inherently limited by the size of the embedding pool T in each MINIconv layer. This value must be predetermined during model design and cannot be extended dynamically afterward, which may pose challenges in scenarios where the total number of tasks is unknown or incrementally growing over time. Second, although MINIconv introduces no additional computational overhead during inference and remains equivalent to standard convolution in terms of forward computation, its use of hard parameter isolation causes the total number of parameters to scale by a factor of T. While this expansion is the trade-off for achieving interference-free learning across tasks, it may impose memory burdens in resource-constrained environments.

In future work, more flexible or compression-aware embedding strategies may be explored to enhance the scalability and deployability of MINI.

6 Conclusion

In this paper, we propose MINI (Minimalistic Incremental Network for Image Restoration), a novel and lightweight framework designed for task-incremental learning across multiple image restoration tasks. Unlike traditional all-in-one models that suffer from parameter conflict, MINI adopts a hard parameter isolation strategy through the introduction of a simple yet effective module called MINIconv. By leveraging embedding pools instead of dynamic meta-networks, MINIconv achieves task-level parameter decoupling without introducing additional computational overhead. Importantly, MINI is a plug-and-play design that can be seamlessly integrated into a wide range of existing image restoration backbones (e.g., NAFNet, Restormer), enabling them to acquire incremental learning capabilities with minimal modification. Moreover, MINI consistently improves the overall multi-task performance of these baselines, while preserving strong performance on each individual task. To further support robust task adaptation, we introduce a lightweight degradation query module and an embedding-consistency regularization (ECR) strategy, which together enhance MINI's fault tolerance and reliability. Extensive experiments across five diverse image restoration tasks demonstrate that MINI achieves strong task-incremental performance with minimal forgetting, significantly outperforming existing generic continual learning methods. Notably, MINI also retains competitive performance compared to fully joint training baselines, while offering the flexibility of sequential task adaptation. We believe that MINI provides a practical and generalizable solution for continual learning in low-level vision. Future work may explore more dynamic embedding mechanisms, better task discovery under unknown settings, and extensions to transformer-based architectures.

References

[1] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17452–17462, 2022.

[2] Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Jiaqi Ma, Tianheng Cheng, Guoli Wang, Xinggang Wang, Qian Zhang, and Lefei Zhang. Prores: Exploring degradation-aware visual prompt for universal image restoration. *arXiv preprint arXiv:2306.13653*, 2023.

[4] Zilong Li, Yiming Lei, Chenglong Ma, Junping Zhang, and Hongming Shan. Prompt-in-prompt learning for universal image restoration. *arXiv preprint arXiv:2312.05038*, 2023.

[5] Xingyu Jiang, Xiuhui Zhang, Ning Gao, and Yue Deng. When fast fourier transform meets transformer for image restoration. In *European Conference on Computer Vision*, pages 381–402. Springer, 2024.

[6] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Controlling vision-language models for multi-task image restoration. In *International Conference on Learning Representations*, 2024.

[7] Jun Cheng, Dong Liang, and Shan Tan. Transfer clip for generalizable image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25974–25984, 2024.

[8] Xiaogang Xu, Shu Kong, Tao Hu, Zhe Liu, and Hujun Bao. Boosting image restoration via priors from pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2900–2909, 2024.

[9] Yuhao Liu, Zhanghan Ke, Fang Liu, Nanxuan Zhao, and Rynson WH Lau. Diff-plugin: Revitalizing details for diffusion-based low-level tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4197–4208, 2024.

[10] Yuang Ai, Huaibo Huang, Xiaoqiang Zhou, Jiexiang Wang, and Ran He. Multimodal prompt perceiver: Empower adaptiveness generalizability and fidelity for all-in-one image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25432–25444, 2024.

[11] 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.

[12] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

[13] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[14] Jingkai Zhou, Varun Jampani, Zhixiong Pi, Qiong Liu, and Ming-Hsuan Yang. Decoupled dynamic filter networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6647–6656, 2021.

[15] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in neural information processing systems*, 32, 2019.

[16] Ningning Ma, Xiangyu Zhang, Jiawei Huang, and Jian Sun. Weightnet: Revisiting the design space of weight networks. In *European Conference on Computer Vision*, pages 776–792. Springer, 2020.

[17] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[18] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. *Advances in neural information processing systems*, 29, 2016.

[19] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11030–11039, 2020.

[20] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3516–3525, 2020.

[21] Sebastian Flennerhag, Andrei A Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. *arXiv preprint arXiv:1909.00025*, 2019.

[22] 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)*, pages 139–154, 2018.

[23] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5769–5780, 2022.

[24] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European conference on computer vision*, pages 17–33. Springer, 2022.

[25] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022.

[26] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Image restoration with mean-reverting stochastic differential equations. *International Conference on Machine Learning*, 2023.

[27] 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 the IEEE conference on computer vision and pattern recognition*, pages 1357–1366, 2017.

[28] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2019.

[29] Tae Hyun Kim, Seungjun Nah, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Conference on Computer Vision and Pattern Recognition*, pages 1–21. IEEE, 2017.

[30] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018.

[31] Wenhan Yang, Wenjing Wang, Haofeng Huang, Shiqi Wang, and Jiaying Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE Transactions on Image Processing*, 30:2072–2086, 2021.

[32] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[33] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[35] 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*, pages 139–149, 2022.

[36] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pages 631–648. Springer, 2022.

[37] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021.

A Potential societal impact

This work focuses on the development of a minimalistic and general framework for task-incremental learning in image restoration. The proposed method can enhance the adaptability and longevity of vision systems deployed in dynamic real-world environments, such as autonomous driving, surveillance, and medical imaging, by enabling them to incrementally learn new restoration tasks without forgetting previous ones. This contributes to more sustainable and upgradable AI systems.

Since the method is task-agnostic and does not rely on sensitive or personal data, we do not foresee direct negative societal impacts. However, as with many vision enhancement technologies, potential misuse in image manipulation or surveillance scenarios should be considered. We encourage responsible deployment aligned with ethical guidelines and privacy regulations.

B Fully embedded vs. Only MINI-conv

To further validate the conclusions discussed in the Discussion section, we compared the final performance of "embedding all parameter components" and "embedding only the convolutional layers" (i.e., MINI-conv). The results are shown in Table 6.

Table 6: Comparison between fully-embedded and only-MINIconv paradigm.

methods	ave PSNR↑	ave SSIM↑	ave LPIPS↓
Fully embedded	28.844	0.915	0.0629
Only MINI-conv	28.754	0.911	0.632

It can be seen that, the average performance improvement was only around 0.1 dB in PSNR. We believe this is because these parameters primarily perform affine and scaling transformations among features, and such transformations tend to exhibit limited variation across tasks within the same network architecture. Therefore, even without explicit parameter isolation, these components do not lead to severe catastrophic forgetting.

C MINI-based transformer architecture

To preliminarily evaluate the performance of the MINI architecture on transformer-based models, we conducted the same experimental tests on SwinIR[37], specifically, we embedded the parameter components within each transformer block to replace the original parameter components. and the results are shown in Table 7. The results indicate that our MINI architecture can also be applied to transformer-based backbone networks to enhance incremental learning capability. However, it is important to ensure that the parameter initialization of the embedding pool remains consistent with that of the original parameter components.

Table 7: Final performance comparison of model-agnostic generic incremental learning methods under SwinIR[37] backbone. The training task sequence is: rain $\rightarrow$ haze $\rightarrow$ blur $\rightarrow$ raindrop $\rightarrow$ low light. The best results are highlighted in **bold**.

datasets	R100H (rain)			RESIDE-6k (haze)			GoPro (blur)			Raindrop (raindrop)			LOLv2 (low light)		
methods / metrics	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
LwF[13]	14.01	0.601	0.636	19.21	0.598	0.263	23.96	0.623	0.189	27.04	0.816	0.095	22.05	0.873	0.105
EWC[11]	14.55	0.563	0.525	20.00	0.611	0.233	20.62	0.634	0.249	24.67	0.789	0.128	**22.83**	0.880	0.100
MINI	**31.23**	**0.921**	**0.040**	**30.41**	**0.961**	**0.026**	**30.65**	**0.901**	**0.089**	**30.57**	**0.911**	**0.054**	22.72	**0.888**	**0.097**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In Section 1 and Abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Described in Section 3.1, 3.2 and 3.3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code will be made publicly available after the camera-ready stage.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: In Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We comply with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: In Appendix A.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The methods and datasets mentioned in this paper are all publicly available and open-source. They are properly cited in the article, and there is no plagiarism or misuse.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We have provided a detailed description of how to use our code in and commit to making our code publicly available on GitHub after the camera-ready stage.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.