RainbowPrompt: Diversity-Enhanced Prompt-Evolving for Continual Learning

Kiseong Hong
Department of AI
Chung-Ang University
ghdrltjd@cau.ac.kr

Gyeong-hyeon Kim
School of CSE
Chung-Ang University
leonardkkh@cau.ac.kr

Eunwoo Kim*
School of CSE
Chung-Ang University
eunwoo@cau.ac.kr

Abstract

Prompt-based continual learning provides a rehearsal-free solution by tuning small sets of parameters while keeping pre-trained models frozen. To meet the complex demands of sequential tasks, it is crucial to integrate task-specific knowledge within prompts effectively. However, existing works rely on either fixed learned prompts (i.e., prompts whose representations remain unchanged during new task learning) or on prompts generated from an entangled task-shared space, limiting the representational diversity of the integrated prompt. To address this issue, we propose a novel prompt-evolving mechanism to adaptively aggregate base prompts (i.e., task-specific prompts) into a unified prompt while ensuring diversity. By transforming and aligning base prompts, both previously learned and newly introduced, our approach continuously evolves accumulated knowledge to facilitate learning new tasks. We further introduce a learnable probabilistic gate that adaptively determines which layers to activate during the evolution process. We validate our method on image classification and video action recognition tasks in class-incremental learning, achieving average gains of 9.07% and 7.40% over existing methods across all scenarios.

1. Introduction

Continual learning (CL) aims to enable a model to learn sequential tasks while retaining knowledge from previous tasks. However, sequential learning faces a major challenge, catastrophic forgetting [9], where learning new tasks deteriorates performance on previously learned tasks. To address this, various CL approaches [20, 22, 36] have been proposed. A recent paradigm in CL, known as prompt-based CL (PCL) [33, 45], has shown promise in mitigating catastrophic forgetting with its rehearsal-free nature, which applied to a pre-trained vision transformer (ViT) [8]. These methods leverage prompt-tuning [25], a transfer learning

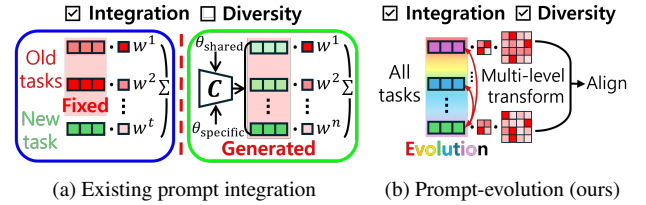---

*Corresponding author.

Figure 1. Conceptual illustration of (a) existing prompt integration approaches and (b) the proposed prompt-evolving approach. (a) Existing approaches integrate prompts using input-conditioned weights w, with fixed old task prompts (blue box) or those generated between task-shared and task-specific spaces (green box). (b) Our approach progressively transforms and aligns prompts to make their representations more adaptable to new tasks.

strategy initially developed for natural language processing (NLP). Instead of modifying the weights of ViT, prompt-tuning keeps the model frozen and fine-tunes additional small sets of parameters called prompts. These prompts act as task-specific instructions, guiding the model without the need for direct weight updates.

Existing PCL works either select task-relevant prompts from a prompt pool [45] or extend this approach by incorporating a task-invariant prompt shared across all tasks [44]. Recent studies attempt to integrate prompts into a unified prompt [35, 38]. They produce a prompt by combining prompts via an input-conditioned weighted sum [38] or by generating task-specific prompts and merging them [35] (see Fig. 1a). However, they often yield prompts with limited representational diversity. This limitation arises from relying on learned representations that remain unchanged without adapting to new tasks [38] or deriving prompts in a task-shared space vulnerable to task interference and dominance [35]. This reduces adaptability and generalization, highlighting the need for more effective strategies to balance integration with diversity.

Furthermore, our findings in Fig. 2 reveal that this limitation hinders achieving high accuracy while minimizing forgetting during sequential task learning. We observe that existing methods fail to achieve optimal knowledge integra-
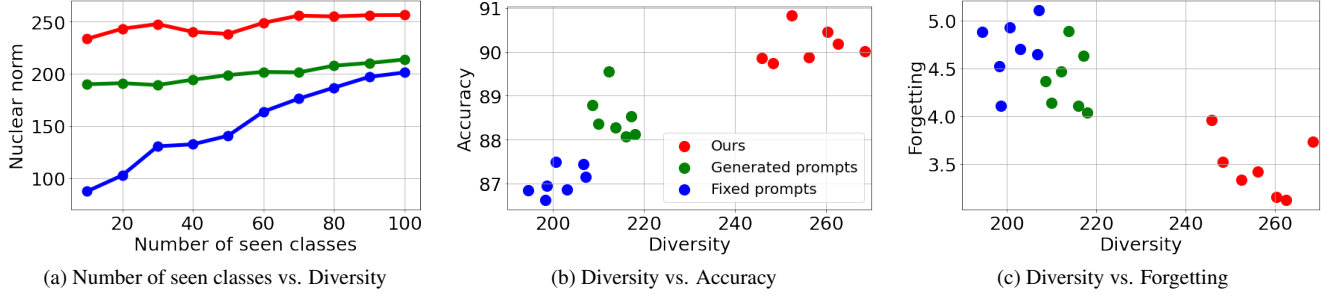
Figure 2. Results of the representation diversity in a class-incremental learning scenario with a 10-task setting on CIFAR-100. We compare ours with two baselines using fixed [38] and generated [35] prompts. In (a), we measure the average nuclear norm of the prompt at each time step as new classes are introduced, using test samples from all seen classes. We insert prompts into all layers and calculate diversity using the prompt in the last layer. In (b) and (c), we compare the representation diversity (after all classes are learned) against accuracy and forgetting, respectively, with each dot denoting a distinct random trial.

tion with enhanced diversity[1].

Merging prompts via a weighted sum without adequate adaptation to new tasks yields low representational diversity (blue line in Fig. 2a). This limitation restricts the ability to capture distinctive features introduced by new classes. Generating prompts for new classes improves representational diversity, as it adapts to class-specific traits (green line in Fig. 2a). However, the diversity of prompts generated from the increasingly entangled task-shared space remains insufficient, as overlapping knowledge across tasks leads to less distinctive representations. Figures 2b and 2c demonstrate that higher representation diversity improves accuracy and reduces forgetting, respectively. These analyses underscore the importance of optimal knowledge integration to ensure diversity and preserve previously learned knowledge in sequential task learning.

In this work, we propose a prompt-evolving approach to integrate task-specific knowledge into a diversity-enhanced representation. Prompts undergo transformation and alignment, gradually adapting their representations to new tasks while preserving previously learned information. Our method progressively reconfigures the prompts to seamlessly integrate complementary knowledge from both previous and current tasks. It consists of two steps: attention-based transformation and task-guided alignment. Our attention-based transformation assesses the relevance of multiple prompts to new tasks. Unlike existing works that derive prompt weights via cosine similarity between input-conditioned queries and prompt keys [38], our method dynamically reweights the contribution of each prompt based on its relevance to new tasks at multi-level granularity. The task-guided alignment further refines these representations by progressively tailoring them to a new task. It incor-

porates non-linear transformations to align the representations of the new task while preserving the intrinsic properties of each prompt. Finally, we introduce RainbowPrompt, a novel prompt that integrates enhanced representations of task-specific prompts. We further optimize the proposed method by introducing a learnable probabilistic gate that adaptively regulates layer activation during evolution, leveraging task-specific differences.

To demonstrate the effectiveness of RainbowPrompt, we compare it with existing prompt-based CL approaches [11, 33, 35, 38, 44, 45] in class-incremental learning scenarios. The evaluation includes image classification tasks using ImageNet-R [14], CIFAR-100 [24], and CUBS [42]. We also evaluate video action recognition tasks on UCF-101 [39] and ActivityNet [2]. Extensive experiments show that the proposed method significantly outperforms the compared methods across all benchmarks, demonstrating its effectiveness and versatility in image and video tasks. The main contributions of this work are as follows:

- We propose a novel prompt-evolving approach for CL, enabling effective knowledge integration while ensuring diversity across tasks.
- Our method employs attention-based transformation and task-guided alignment to integrate knowledge from previous tasks while adapting to new tasks.
- Experimental results on a wide range of tasks demonstrate that ours outperforms its competitors with an overall average margin of 8.23% across all scenarios.

2. Related Work

Continual Learning. Continual learning (CL) aims to progressively acquire new knowledge while retaining previously learned information [7, 15]. A major challenge in CL is catastrophic forgetting [9], where learning new tasks degrades performance on earlier ones. Solutions to this challenge can be broadly divided into three categories:

---

[1]We analyze the nuclear norm of the prompt as a measure of representation diversity. A higher nuclear norm reflects greater representation diversity and improves the discriminability between classes [6, 13].

Regularization-based methods [1, 22] mitigate catastrophic forgetting by adding penalty terms that consolidate knowledge from earlier tasks. Rehearsal-based methods tackle catastrophic forgetting by retaining a subset of the data from previous tasks [4, 29, 34] or generating data for previous tasks using adversarial techniques [32, 37, 46]. Architecture-based methods [17, 19, 47, 48] dynamically adjust the network structure to accommodate new tasks. Despite the advancements of these approaches, the emergence of foundation models trained on large-scale data has driven a paradigm shift in CL [45], moving towards replay-free and parameter-efficient approaches [21, 27].

Prompt-based Continual Learning. Prompting [28] initially involved manually designed task-specific instructions to induce desired responses from pre-trained models. In the context of CL, prompt-based methods [33, 45] retain task-specific knowledge without rehearsal buffers. They use prompts as keys to retrieve stored knowledge, eliminating the need for model parameter updates. L2P [45] introduces prompt-tuning with key-query matching to retrieve task-relevant knowledge. DualPrompt [44] further introduces task-invariant prompts along with task-specific prompts to distinctly encode task-relevant and task-agnostic knowledge. CODA-Prompt [38] introduces a decomposed attention-based prompting method, while ConvPrompt [35] leverages convolution for prompt generation. PGP constrains prompt updates to be orthogonal to prior prompt directions via a projection mechanism. CPrompt [11] introduces consistency prompting to address inconsistencies between the classifier and the prompt.

Limitations. Among the competitive approaches, CODA-Prompt [38] and ConvPrompt [35] focus on integrating multiple base prompts but often overlook diversity within the resulting unified prompt. Fixed representations hinder the capture of diverse information [3, 5], while generated representations shaped by a task-shared space often suffer from task interference and dominance, resulting in overfitting and limited generalization [23, 30]. In contrast, we address these limitations by progressively evolving the representations of all accumulated base prompts. We accumulate independent base prompts for each task and integrate them into a unified prompt for optimal knowledge integration.

3. Methodology

3.1. Prerequisites

Continual learning (CL) enables a model to progressively learn knowledge from a sequence of tasks $\{\mathcal{T}^1, \mathcal{T}^2, \ldots, \mathcal{T}^T\}$. The t-th task $\mathcal{T}^t$ consists of N_t input-label pairs $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N_t}$, where $\boldsymbol{x}_i \in \mathcal{X}^t$ represents the input data and $y_i \in \mathcal{Y}^t$ denotes the corresponding label. In this work, we focus on class-incremental learning, where task identities are unknown during testing [16]. We also adopt a practical rehearsal-free setting [44], which prohibits the storage of data from previous tasks.

In prompt-based continual learning (PCL), the model $f_\theta(\cdot)$ is typically chosen to be a pre-trained vision transformer (ViT) [8]. Rather than adjusting the weights of ViT directly, we use small learnable parameters, i.e., prompts $\boldsymbol{p} \in \mathbb{R}^{L_p \times D}$, to provide task-specific instructions [25], where L_p represents the prompt length and D is the embedding dimension. Specifically, we employ prefix tuning (P-T) [26], where the prompts comprising $\boldsymbol{p}_K \in \mathbb{R}^{L_p/2 \times D}$ and $\boldsymbol{p}_V \in \mathbb{R}^{L_p/2 \times D}$ are prepended to the key and value representations in the multi-head self-attention layers of ViT.

3.2. Prompt-Evolving Mechanism

Our goal is to consolidate task-specific prompts into a cohesive prompt through a prompt-evolving mechanism, which dynamically reconfigures their representations to adapt to new tasks while enhancing representational diversity. At each time step t, a new task is introduced, with its corresponding base prompts (i.e., task-specific prompts) $\boldsymbol{p}^t = \{\boldsymbol{p}_l^t\}_{l=1}^L$, where $\boldsymbol{p}_l^t \in \mathbb{R}^{L_p \times D}$ is the base prompt at layer l, and L is the number of layers. To facilitate sequential learning, we maintain a set of accumulated prompts denoted as $\mathcal{P} = \{\boldsymbol{p}^1, \boldsymbol{p}^2, \ldots, \boldsymbol{p}^t\}$, which includes all base prompts up to time step t. Our approach seeks to integrate these prompts into unified prompts, i.e., $\boldsymbol{p}^{\text{rainbow}(t)} = \{\boldsymbol{p}_l^{\text{rainbow}(t)}\}_{l=1}^L$, where $\boldsymbol{p}_l^{\text{rainbow}(t)}$ is the unified prompt at layer l, without hurting the base prompts.

The accumulated prompts $\mathcal{P}$ present a challenge when combining them in continual learning. Direct integration of prompts risks losing knowledge from previously learned tasks, while freezing them restricts adaptation and hinders knowledge transfer to the t-th task. To address this, we update the base prompts for the current task at each time step while keeping the base prompts from previous tasks frozen to preserve their knowledge. To facilitate knowledge transfer to new tasks, we transform and align the representations of all base prompts by introducing learnable components. Specifically, we apply self-attention to dynamically reweight the contributions of accumulated prompts, facilitating their transformation by emphasizing salient information while suppressing less relevant details. This adaptive reweighting mitigates knowledge dilution and enhances representational diversity. By aligning the prompts to their transformed representations for the new task, we encapsulate the diverse knowledge in $\mathcal{P}$ into a set of unified prompts. The overall framework is illustrated in Fig. 3.

3.2.1. Attention-based Transformation

We first apply an attention-based transformation that enables accumulated prompts to self-identify their influence on the new task via two-level interactions: task-level and feature-level. Interactions among prompt vectors capture
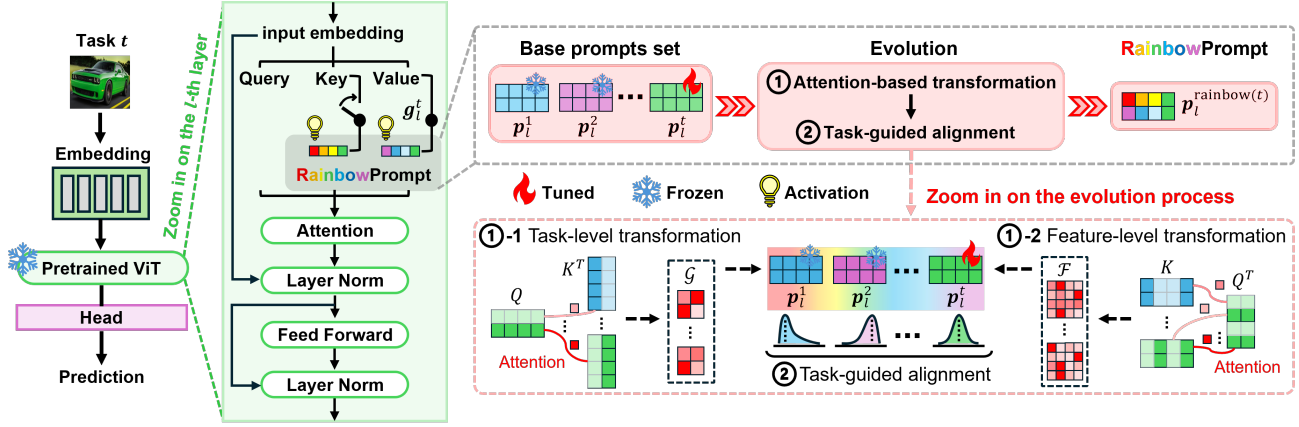
Figure 3. Illustration of the proposed framework. When a new task t arrives, we evolve the base prompt representations, comprising all task-specific prompts up to the t-th time step, through attention-based transformation and task-guided alignment. Finally, RainbowPrompt, $\boldsymbol{p}_l^{\text{rainbow}(t)}$, is constructed at each layer l by integrating diverse knowledge from the accumulated base prompts. We further introduce a learnable probabilistic gate $\boldsymbol{g}_l^t$ to selectively insert $\boldsymbol{p}_l^{\text{rainbow}(t)}$ into intermediate layers. During testing, the proposed method requires the RainbowPrompts $\{\boldsymbol{p}_l^{\text{rainbow}(i)}\}_{l=1}^{L_i}$ for each task i, which are appended to L_i selected layers determined by $\{\boldsymbol{g}_l^i\}_{l=1}^{L}$.

overall contributions at the task level, while interactions among the individual features (elements) of each vector measure fine-grained contributions. Before the transformation process, we precede a task-conditioning step to inject task-relevant information into $\mathcal{P}_l$[2].

We use a learnable task embedding vector $\boldsymbol{e}^t \in \mathbb{R}^D$ to compute attention weights that emphasize the task-relevant components of $\mathcal{P}_l$ at the t-th time step. We condition the set of base prompts by $\mathcal{P}_l \leftarrow \text{softmax}\big(\sigma(\boldsymbol{e}^t)\mathcal{P}_l^T/\sqrt{d_p}\big)\mathcal{P}_l$, where $\sigma(\cdot)$ and d_p denote the broadcasting function and the dimensionality of the $\mathcal{P}_l$, respectively.

We treat $\boldsymbol{p}_l^{\text{new}} = \boldsymbol{p}_l^t$ as the query and use the concatenated task-conditioned base prompts set $\mathcal{P}_l$ as the key and value inputs to the transformation process. First, we project each prompt into a lower-dimensional space to align their dimensions for the transformation, reducing the complexity of subsequent operations:

$$Q = W_l^Q \boldsymbol{p}_l^{\text{new}}, \ K = W_l^K \mathcal{P}_l, \ V = W_l^V \mathcal{P}_l, \quad (1)$$

where $W_l^Q \in \mathbb{R}^{D \times D_p}$, $W_l^K \in \mathbb{R}^{D \times D_p}$, and $W_l^V \in \mathbb{R}^{D \times D_p}$ are learnable projection matrices for the respective query, key, and value at the l-th layer, and D_p represents the projected dimension ($D_p \ll D$).

For the task-level transformation, we define the attention-based transformation function, $\text{AT}(\cdot)$, which computes the inter-task affinity matrix $\mathcal{G}$ from the attention between Q and K, and then uses $\mathcal{G}$ to weight the value representations:

$$\text{AT}(Q, K, V) = \mathcal{G} \cdot V \triangleq \tilde{V} \in \mathbb{R}^{t \times L_p \times D_p}, \quad (2)$$

---

[2]We describe the method at a specific layer l for notational convenience.

where $\mathcal{G} = \text{softmax}\big(QK^T/\sqrt{d_k}\big)$, d_k denotes the dimensionality of the key, $\cdot$ represents the matrix multiplication operation, and $\tilde{V}$ is the transformed representation. $\mathcal{G}$ quantifies how much information from existing tasks should contribute to the new task, allowing us to adaptively weigh the influence of each task. By integrating $\mathcal{G}$ with V, we transform the representations associated with the prompts of each task based on their contribution to the new task t.

The feature-level transformation captures cross-feature influences at a finer granularity, complementing the task-level transformation. It leverages transposed query Q^T and key K to enable comparison of feature dimensions. This approach is inspired by bilinear pooling [10], which enjoys multiplicative interactions between feature dimensions. We update the output of the task-level transformation $\tilde{V}$ as

$$\text{AT}(Q, K, \tilde{V}) = \mathcal{F} \cdot \tilde{V}^T \triangleq \hat{V} \in \mathbb{R}^{t \times D_p \times L_p}, \quad (3)$$

where the inter-feature affinity matrix $\mathcal{F}$ is computed as $\mathcal{F} = \text{softmax}\big(Q^T K/\sqrt{d_k}\big)$. It quantifies the contributions between individual features across tasks, capturing fine-grained dependencies. By applying $\mathcal{F}$ to $\tilde{V}$, the feature-level transformation integrates cross-feature influences, refining $\tilde{V}$ to incorporate contributions distinct from those of the task-level transformation.

To integrate the transformed representations, we redefine the output of the feature-level transformation as $\hat{V} \leftarrow \text{LN}(\mathcal{P}_l + (\hat{V}^T W_l^O)) \in \mathbb{R}^{t \times L_p \times D}$. $\text{LN}(\cdot)$ represents layer normalization, and $W_l^O \in \mathbb{R}^{D_p \times D}$ is a learnable projection matrix. This integrates $\mathcal{P}_l$ with the transformed representations, preserving the original information in $\mathcal{P}_l$ while enriching the resulting representations with newly adapted features.

3.2.2. Task-Guided Alignment

Following the attention-based transformation, the task-guided alignment refines the transformed representations $\hat{V}$. This decodes the representations to meet the traits of the new task, preserving the attributes of each prompt:

$$\tilde{\mathcal{P}}_l = \text{LN}(\hat{V} + \text{LT}(\hat{V})) \in \mathbb{R}^{t \times L_p \times D}, \qquad (4)$$

where $\text{LT}(x) = \max(0, xW_l^1)W_l^2$ aligns the transformed representations to identify task-relevant patterns. It refines their distribution across tasks within a reduced space D_n ($D_n \ll D$) [12, 31], with $W_l^1 \in \mathbb{R}^{D \times D_n}$ and $W_l^2 \in \mathbb{R}^{D_n \times D}$ as learnable weight matrices (bias terms omitted for simplicity).

Finally, we derive a RainbowPrompt, $\boldsymbol{p}_l^{\text{rainbow}(t)}$, by combining the evolved representations of prompts from all tasks learned up to time step t into a unified prompt:

$$\boldsymbol{p}_l^{\text{rainbow}(t)} = \frac{1}{t} \sum_{i=1}^{t} \tilde{\mathcal{P}}_l[i] \in \mathbb{R}^{L_p \times D}, \qquad (5)$$

where $\tilde{\mathcal{P}}_l[i]$ denotes the evolved representation of the i-th base prompt. $\boldsymbol{p}_l^{\text{rainbow}(t)}$ is a distinct type of prompt that differs from task-specific and task-shared prompts by evolving and integrating accumulated knowledge without predefined roles. Note that the proposed prompt-evolving mechanism is performed exclusively during training, where task-specific RainbowPrompts are produced and stored. At test time, they are directly used for prediction without requiring the evolution components.

3.3. Adaptive Prompting

Determining which layers to insert RainbowPrompts poses an additional challenge due to varying learning complexities across tasks. Manually selecting these layers is impractical, as it fails to account for task-specific differences [15]. To this end, we introduce a task-specific learnable probabilistic gate $\boldsymbol{G}^t = \{\boldsymbol{g}_l^t\}_{l=1}^L$ to learn where to insert RainbowPrompts in the model. $\boldsymbol{g}_l^t$ is a Bernoulli random variable deciding whether to insert a Rainbow-Prompt at layer l. Since the discrete nature of $\boldsymbol{g}_l^t$ prevents gradient-based optimization, we employ the Gumbel-Softmax trick [18] to relax it into a differentiable form:

$$\hat{g}_l^t(p) = \frac{\exp\left(\log \delta_l^t(p) + Z_l^t(p)/\tau\right)}{\sum_{i \in \{0,1\}} \exp\left(\log \delta_l^t(i) + Z_l^t(i)/\tau\right)}, \qquad (6)$$

where $p \in \{0, 1\}$, $Z_l^t = -\log(-\log U_l^t)$ is the Gumbel noise generated from $U_l^t \sim \mathcal{U}[0,1]$, and τ controls the temperature for continuous relaxation. We sample discrete task-specific decisions from the learned distribution $\delta_l^t = [\alpha_l^t, 1 - \alpha_l^t]$, where α_l^t is the probability of prompt insertion for layer l. This flexible activation aligns more effectively with adaptive prompting, compared to the manual

prompting [35, 38]. Section 4.4 provides a detailed discussion on this aspect.

3.4. Optimization

We jointly optimize δ_l^t, the new task prompt $\boldsymbol{p}_l^t$, and the learnable parameters of the prompt-evolving mechanism, defined as $W^{\text{evolution}} = \{(W_l^Q, W_l^K, W_l^V, W_l^O, W_l^1, W_l^2)\}_{l=1}^L$, using the cross-entropy loss. To encourage a sparse yet effective RainbowPrompt insertion pattern, we impose a regularization term to suppress insertion probabilities α_l^t, i.e., $\mathcal{L}_{\text{sparse}} = \sum_{l \leq L} \log \alpha_l^t$. In addition, we optimize the task embedding vector $\boldsymbol{e}^t$ using a matching loss [44], $\mathcal{L}_{\text{match}} = \gamma(q(\boldsymbol{x}), \boldsymbol{e}^t)$, where $\boldsymbol{x} \in \mathcal{T}^t$, γ denotes cosine similarity, and $q(\cdot)$ is a query function [44]. Finally, the proposed method minimizes the following total loss:

$$\min_{\Theta^t} \sum_{i=1} \text{CE}(\boldsymbol{z}_i, y_i) + \lambda_s \mathcal{L}_{\text{sparse}} + \lambda_m \mathcal{L}_{\text{match}}, \qquad (7)$$

where $\Theta^t = \{\boldsymbol{p}^t, \boldsymbol{e}^t, \boldsymbol{G}^t, W^{\text{evolution}}, \phi\}$, where ϕ denotes a classifier. CE is the cross-entropy loss, $\boldsymbol{z}_i$ represents the final output, and λ_s and λ_m are balancing parameters, both set to 0.01 in our experiments.

4. Experiments

4.1. Setup

Scenarios and Datasets. To evaluate RainbowPrompt[3], we demonstrated it on image classification and video action recognition tasks under class-incremental learning scenarios. For image classification, we used the ImageNet-R [14], CIFAR-100 [24], and CUBS [42] datasets. ImageNet-R is a challenging benchmark due to its classes featuring distinct styles (e.g., cartoon, graffiti, origami) and significant intra-class diversity [44]. We divided its 200 classes into disjoint subsets, forming 10-task (20 classes per task) and 20-task (10 classes per task) settings. CIFAR-100, a widely used benchmark in prompt-based continual learning (PCL), was split into 10-task (10 classes per task) and 20-task (5 classes per task) settings. CUBS, a fine-grained dataset featuring bird species, poses another challenge due to subtle inter-class differences. We applied the same task split configuration as used for ImageNet-R. When splitting each dataset, we ensure no class overlaps occur between them.

For video action recognition, we conducted experiments using the vCLIMB benchmark [40] on trimmed versions of the UCF-101 [39] and ActivityNet [2] datasets. We evaluated our approach on UCF-101 (101 classes) and ActivityNet (200 classes) under both 10-task and 20-task settings. Following [40], we partitioned each dataset into disjoint subsets of randomly selected classes, with each subset treated as an independent task.

---

[3]We also represent RainbowPrompt as our method.

Table 1. Results on ImageNet-R and CIFAR-100. A_N and F_N represent the average accuracy and forgetting for the N-task setting, respectively. The best performance is highlighted in bold.

Method	ImageNet-R				CIFAR-100			
	A_{10} (↑)	F_{10} (↓)	A_{20} (↑)	F_{20} (↓)	A_{10} (↑)	F_{10} (↓)	A_{20} (↑)	F_{20} (↓)
Joint training	79.60 ± 0.87	-	79.60 ± 0.87	-	93.22 ± 0.16	-	93.22 ± 0.16	-
L2P [45]	63.49 ± 0.40	6.85 ± 0.42	59.38 ± 0.50	5.89 ± 0.36	82.76 ± 1.17	7.86 ± 0.39	77.95 ± 0.72	9.88 ± 0.30
DualPrompt [44]	68.50 ± 0.52	5.14 ± 0.18	63.21 ± 0.49	5.28 ± 0.45	85.07 ± 0.49	5.57 ± 0.20	80.49 ± 0.31	8.84 ± 0.68
DualPrompt-PGP [33]	69.34 ± 0.05	4.53 ± 0.04	64.75 ± 0.38	6.04 ± 0.15	86.92 ± 0.05	5.35 ± 0.19	83.74 ± 0.01	7.91 ± 0.15
CODA-Prompt [38]	74.24 ± 0.56	4.92 ± 0.21	70.86 ± 0.42	6.87 ± 0.25	87.00 ± 0.38	4.78 ± 0.24	82.15 ± 0.17	6.33 ± 0.23
CPrompt [11]	76.71 ± 0.61	4.66 ± 0.93	74.45 ± 0.25	4.98 ± 0.21	87.83 ± 0.37	4.88 ± 0.79	84.66 ± 0.13	5.69 ± 0.06
ConvPrompt [35]	77.86 ± 0.25	4.33 ± 0.24	75.10 ± 0.39	4.10 ± 0.29	88.87 ± 0.33	4.75 ± 0.15	87.37 ± 0.13	5.16 ± 0.01
RainbowPrompt (ours)	$\mathbf{79.09 \pm 0.13}$	$\mathbf{3.90 \pm 0.23}$	$\mathbf{78.36 \pm 0.47}$	$\mathbf{3.44 \pm 0.29}$	$\mathbf{89.86 \pm 0.11}$	$\mathbf{3.44 \pm 0.26}$	$\mathbf{90.15 \pm 0.05}$	$\mathbf{3.75 \pm 0.23}$

Compared Methods. We evaluated RainbowPrompt against existing PCL approaches: L2P [45], Dual-Prompt [44], DualPrompt-PGP [33], CODA-Prompt [38], CPrompt [11], and ConvPrompt [35]. We also reported the results of joint training, where the ViT model [8] was trained on a combined dataset encompassing the training data from all tasks. We evaluated the methods using two metrics: average accuracy and average forgetting [45].

Implementation Details. We followed the training details from [38], including the optimizer and input resolution, while using a learning rate of 0.03. We used a single base prompt with a prompt length of $L_p = 20$ per task across all scenarios in RainbowPrompt. We set the dimensionality of the projection matrices (D_p) to 96 for the 10-task setting and 56 for the 20-task setting. We set the dimensionality of the weight matrices (D_n) as 56 for the 10-task setting and 28 for the 20-task setting. LT$(\cdot)$ consists of two linear layers stacked with ReLU activations. For adaptive prompting, we used soft decisions from the distribution δ_l^t during training for a certain number of epochs. After that, we sampled task-specific decisions from the learned distribution. We implemented all methods, including RainbowPrompt, using the pre-trained ViT-B/16 as in [45]. We conducted each experiment three times with random trials and reported the results as the mean and standard deviation of the runs.

4.2. Image Classification

We first evaluated the proposed approach on image classification using the ImageNet-R dataset. Tab. 1 (left) presents the results for both 10-task and 20-task settings. In the 10-task setting, RainbowPrompt surpasses all the competitors, L2P, DualPrompt, DualPrompt-PGP, CODA-Prompt, CPrompt, and ConvPrompt, in average accuracy by gaps of 15.60%, 10.59%, 9.75%, 4.85%, 2.38%, and 1.23%, respectively. Compared to ConvPrompt, which achieves the lowest average forgetting among the compared methods, RainbowPrompt further reduces forgetting by 0.43%, highlighting its superior capability to retain knowledge while accommodating new tasks. The results remain consistent in the more challenging 20-task setting. Notably, Rainbow-

Prompt achieves an accuracy improvement of 3.26% and a forgetting reduction of 0.66% compared to the strong competitor, ConvPrompt. These results highlight the effectiveness of ours in both the 10-task and 20-task settings.

We further evaluated RainbowPrompt on CIFAR-100. The results for the 10-task and 20-task settings are summarized in Tab. 1 (right). Notably, in the 20-task setting, L2P, DualPrompt, DualPrompt-PGP, CODA-Prompt, CPrompt, and ConvPrompt exhibit accuracy drops of 4.81%, 4.58%, 3.18%, 4.85%, 3.17%, and 1.50%, respectively, compared to their performance in the 10-task setting. In contrast, RainbowPrompt not only avoids performance degradation but achieves a 0.29% increase in accuracy on average. It demonstrates consistently competitive performance as the number of tasks increases, while others struggle to adapt under extended task sequences.

We evaluated RainbowPrompt on the fine-grained CUBS dataset. Fig. 4 illustrates the average accuracy and forgetting across all seen classes. RainbowPrompt outperforms all baseline methods in terms of the metrics. In the 10-task setting, RainbowPrompt and the strongest baseline, ConvPrompt, exhibit stable performance as the number of seen classes increases. RainbowPrompt achieves a 3.62% higher accuracy and 2.28% lower forgetting than ConvPrompt. Notably, in the 20-task setting, Rainbow-Prompt significantly outperforms ConvPrompt, achieving a notable 10.44% improvement in average accuracy and a 5.44% reduction in forgetting. It demonstrates the ability to produce diversity-enhanced prompts while accommodating new tasks.

4.3. Video Action Recognition

To extend the evaluation of the proposed prompt-evolving mechanism beyond image classification, we applied it to video action recognition tasks. We utilized trimmed versions of the UCF-101 and ActivityNet datasets [41] to focus on well-defined actions and facilitate analysis by excluding irrelevant background segments. Each video was divided into three equal segments, and one frame was randomly sampled from each segment [43]. The datasets
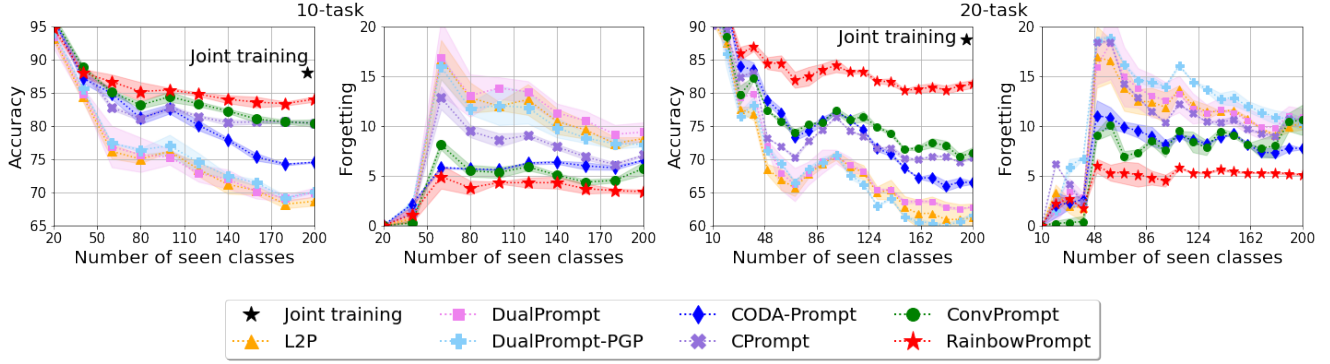
Figure 4. Results on CUBS. The average accuracy for all seen classes at each time step is shown with its standard deviation indicated by the shaded area.

Table 2. Results on the video action recognition tasks using UCF-101 and ActivityNet.

Method	UCF-101				ActivityNet			
	A_{10} (↑)	F_{10} (↓)	A_{20} (↑)	F_{20} (↓)	A_{10} (↑)	F_{10} (↓)	A_{20} (↑)	F_{20} (↓)
L2P [45]	78.35 ± 1.54	5.46 ± 0.29	70.29 ± 0.19	8.31 ± 0.40	63.46 ± 0.47	6.47 ± 1.13	56.47 ± 1.12	9.96 ± 1.33
DualPrompt [44]	83.15 ± 0.82	5.41 ± 0.38	74.96 ± 1.23	7.97 ± 0.41	63.76 ± 0.88	5.78 ± 0.50	56.59 ± 0.81	9.89 ± 0.05
DualPrompt-PGP [33]	84.39 ± 0.56	5.92 ± 0.16	75.37 ± 0.16	7.50 ± 0.59	64.65 ± 0.36	6.20 ± 0.64	57.46 ± 0.64	8.56 ± 0.25
CODA-Prompt [38]	84.77 ± 0.75	5.81 ± 0.27	75.35 ± 0.90	5.66 ± 0.50	66.13 ± 0.65	6.23 ± 0.15	58.62 ± 0.58	8.27 ± 0.34
CPrompt [11]	87.16 ± 0.59	5.30 ± 0.25	81.78 ± 0.04	4.42 ± 0.19	66.81 ± 0.48	7.25 ± 0.08	62.17 ± 0.66	7.49 ± 0.57
ConvPrompt [35]	85.58 ± 0.50	5.34 ± 0.25	78.83 ± 0.29	4.03 ± 0.79	67.32 ± 0.30	5.04 ± 0.29	60.01 ± 0.34	6.02 ± 0.52
RainbowPrompt (ours)	$\mathbf{89.03 \pm 0.91}$	$\mathbf{4.91 \pm 0.30}$	$\mathbf{84.05 \pm 1.27}$	$\mathbf{3.59 \pm 0.17}$	$\mathbf{69.87 \pm 0.24}$	$\mathbf{3.96 \pm 0.28}$	$\mathbf{70.55 \pm 1.57}$	$\mathbf{5.47 \pm 0.47}$

were prepared using the temporal segment sampling strategy outlined in [43]. We evaluated the proposed approach on the 10-task and 20-task settings for each dataset, with the results presented in Tab. 2.

In the 10-task setting on UCF-101, RainbowPrompt outperforms L2P, DualPrompt, DualPrompt-PGP, CODA-Prompt, CPrompt, and ConvPrompt in average accuracy by 10.68%, 5.88%, 4.64%, 4.26%, 1.87%, and 3.45%, respectively. Also, it reduces forgetting by 0.55%, 0.50%, 1.01%, 0.90%, 0.39%, and 0.43% compared to these methods. In the 20-task setting, it outperforms all competing methods by a larger gap than in the 10-task setting. Specifically, it achieves an average improvement of 7.95% in accuracy and a 3.92% reduction in forgetting compared to other methods. This is because a larger number of tasks enhances diversity of accumulated knowledge, allowing for a more effective evolution process.

In the 10-task setting on ActivityNet, RainbowPrompt outperforms CPrompt and ConvPrompt by achieving 3.06% higher accuracy and 3.29% lower forgetting than CPrompt and an additional 2.55% accuracy gain and 1.08% lower forgetting over ConvPrompt. In the 20-task setting, it achieves a significant performance gain over ConvPrompt, with a 10.54% improvement in accuracy and a 0.55% reduction in forgetting. Notably, while ConvPrompt experiences a 7.31% drop in accuracy from its 10-task setting, Rainbow-Prompt not only prevents degradation but instead improves

Table 3. Ablation study of RainbowPrompt on ImageNet-R.

TC	TLT	FLT	TGA	AP	A_{10} (↑)	F_{10} (↓)
✓	✓	✓	✓	✓	$\mathbf{79.09 \pm 0.13}$	$\mathbf{3.90 \pm 0.23}$
-	✓	✓	✓	✓	78.92 ± 0.06	4.19 ± 0.14
✓	-	✓	✓	✓	78.70 ± 0.41	4.14 ± 0.00
✓	✓	-	✓	✓	78.57 ± 0.11	4.29 ± 0.36
✓	✓	✓	-	✓	66.31 ± 0.48	4.84 ± 0.26
✓	✓	✓	✓	-	78.13 ± 0.22	4.07 ± 0.19

by 0.68%. The diversity-enhancing integration of Rainbow-Prompt promotes predictive discriminability [6], enabling it to handle complex action classes more effectively.

4.4. Analysis

Ablation Study. We conducted an ablation study on ImageNet-R in the 10-task setting to evaluate the impacts of the key components in RainbowPrompt. Tab. 3 presents the effect of excluding each component, including the task-conditioning step (TC), task-level transformation (TLT), feature-level transformation (FLT), task-guided alignment (TGA), and adaptive prompting (AP). Excluding the task-conditioning step results in small declines in average accuracy and forgetting, suggesting that it provides supplementary task-specific information. Compared to the task-level transformation, which captures overall task dependencies, the feature-level transformation plays a more critical
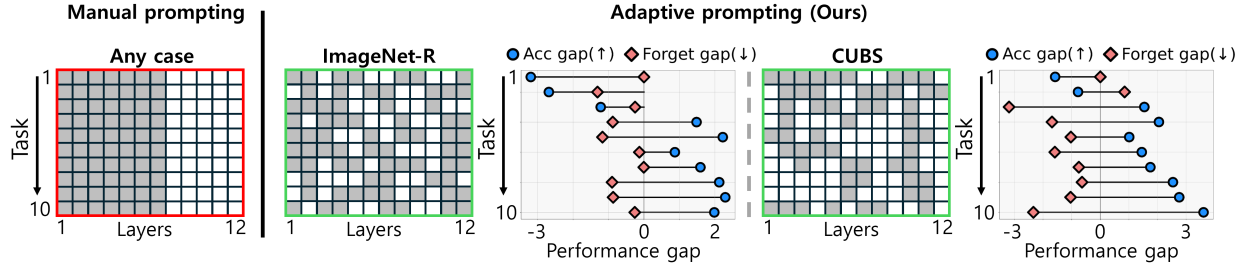
Figure 5. Visualization of prompt insertion patterns across layers: manual prompting (red box) [35] vs. the proposed adaptive prompting (green boxes). Gray represents layers with inserted prompts, whereas white denotes layers without prompts. Accuracy and forgetting gaps denote the average differences in performance across all seen classes, measured as the difference between ours and manual prompting.

role by capturing fine-grained feature interactions, leading to a larger performance drop when removed. Excluding the task-guided alignment leads to the most significant decline in performance, underscoring its essential role in Rainbow-Prompt. It aligns prompts based on their transformed representations, creating a coherent prompt that facilitates new task learning and integrates diverse knowledge. We also observe that replacing manual prompting with an adaptive approach yields performance gains through task-specific tuning strategies.

Adaptive Prompting. We conducted an empirical analysis to demonstrate the advantages of the proposed adaptive prompting, as shown in Fig. 5. We visualize the task-specific prompt insertion layers of manual [35] and adaptive prompting in the 10-task setting using ImageNet-R and CUBS. Manual prompting applies predetermined prompt-insertion layers uniformly across all tasks [11, 38], regardless of the dataset. In contrast, our adaptive prompting dynamically determines the optimal prompt-insertion layers for each dataset and task. For ImageNet-R, adaptive prompting outperforms manual prompting, achieving a 1.98% accuracy gain and 0.26% less forgetting across all tasks. We observe a drop in accuracy in the first three tasks, while forgetting remains consistently lower or comparable to manual prompting. Note here that integrating historical knowledge is crucial for transferring knowledge to new tasks, but its absence in early sequential learning limits accuracy as shown in the figure. For CUBS, adaptive prompting achieves a 3.61% higher average accuracy and 2.34% lower forgetting than manual prompting. Our approach demonstrates exceptional performance across most tasks, with even larger performance gaps than those observed on ImageNet-R.

Sensitivity. To assess the robustness of our method to hyper-parameter variations, we performed sensitivity analysis on the prompt length (L_p) and the dimensions of the projection and weight matrices (D_p, D_n). As illustrated in Fig. 6, RainbowPrompt consistently outperforms the strong baselines, CODA-Prompt and ConvPrompt, across various
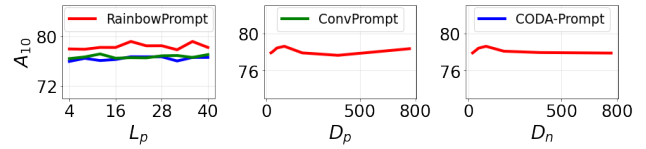

Figure 6. Sensitivity analysis on ImageNet-R.

L_p values (see first figure). Additionally, varying D_p or D_n shows low sensitivity to changes in dimensionality, as shown in the second and third figures. These results indicate that our method is robust to hyper-parameter choices and maintains stable performance.

Efficiency. To evaluate the computational efficiency of our approach, we assessed performance on CIFAR-100 (20-task). CODA-Prompt and ConvPrompt incur 33.7B and 17.1B MACs with 4.8M and 5.7M trainable parameters. Our method trains with 8.2M parameters, including prompt-evolving components, but discards 76.5% (6.2M) at inference, requiring only 18.5B MACs. This offers a favorable balance between accuracy and efficiency.

5. Conclusion

We have proposed RainbowPrompt, a novel prompt-evolving mechanism to address a fundamental challenge in sequential task learning, which effectively integrates accumulated historical knowledge without compromising previously learned information. Unlike existing methods, RainbowPrompt progressively improves the representations of task-specific prompts for new tasks to promote integration that enhances diversity. By dynamically regulating layer activation with a learnable probabilistic gate, our method optimizes the evolution process based on task-specific differences. Extensive experiments on image classification and video action recognition benchmarks demonstrate that RainbowPrompt consistently outperforms state-of-the-art methods, achieving significant improvements in accuracy and forgetting across diverse scenarios.

References

[1] 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. 3

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 2, 5

[3] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 9516–9525, 2021. 3

[4] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018. 3

[5] Zhiyuan Chen and Bing Liu. *Lifelong machine learning*. Morgan & Claypool Publishers, 2018. 3

[6] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3941–3950, 2020. 2, 7

[7] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2021. 2

[8] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3, 6

[9] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4): 128–135, 1999. 1, 2

[10] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–326, 2016. 4

[11] Zhanxin Gao, Jun Cen, and Xiaobin Chang. Consistent prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28463–28473, 2024. 2, 3, 6, 7, 8

[12] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020. 5

[13] Dongyoon Han, Sangdoo Yun, Byeongho Heo, and YoungJoon Yoo. Rethinking channel dimensions for efficient model design. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 732–741, 2021. 2

[14] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 2, 5

[15] Kiseong Hong, Hyundong Jin, Sungho Suh, and Eunwoo Kim. Exploration and exploitation in continual learning. *Neural Networks*, 188:107444, 2025. 2, 5

[16] Y Hsu. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018. 3

[17] 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. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 5

[19] Hyundong Jin and Eunwoo Kim. Helpful or harmful: Inter-task association in continual learning. In *European Conference on Computer Vision*, pages 519–535. Springer, 2022. 3

[20] Hyundong Jin, Gyeong-hyeon Kim, Chanho Ahn, and Eunwoo Kim. Growing a brain with sparsity-inducing generation for continual learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 18961–18970, 2023. 1

[21] Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating instance-level prompts for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11847–11857, 2023. 3

[22] Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. Continual learning with node-importance based adaptive group sparse regularization. *Advances in Neural Information Processing Systems*, 33:3647–3658, 2020. 1, 3

[23] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15190–15200, 2023. 3

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 5

[25] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 1, 3

[26] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 3

[27] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024. 3

[28] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 3

[29] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[30] Chengcheng Ma, Yang Liu, Jiankang Deng, Lingxi Xie, Weiming Dong, and Changsheng Xu. Understanding and mitigating overfitting in prompt tuning for vision-language models. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9): 4616–4629, 2023. 3

[31] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022. 5

[32] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11321–11329, 2019. 3

[33] Jingyang Qiao, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, Yuan Xie, et al. Prompt gradient projection for continual learning. In *The Twelfth International Conference on Learning Representations*, 2023. 1, 2, 3, 6, 7

[34] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 3

[35] Anurag Roy, Riddhiman Moulick, Vinay K Verma, Saptarshi Ghosh, and Abir Das. Convolutional prompting meets language models for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23616–23626, 2024. 1, 2, 3, 5, 6, 7, 8

[36] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021. 1

[37] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[38] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023. 1, 2, 3, 5, 6, 7, 8

[39] K Soomro. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 5

[40] Andrés Villa, Kumail Alhamoud, Victor Escorcia, Fabian Caba, Juan León Alcázar, and Bernard Ghanem. vclimb: A novel video class incremental learning benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19035–19044, 2022. 5

[41] Andrés Villa, Juan León Alcázar, Motasem Alfarra, Kumail Alhamoud, Julio Hurtado, Fabian Caba Heilbron, Alvaro Soto, and Bernard Ghanem. Pivot: Prompting for video continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24214–24223, 2023. 6

[42] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 5

[43] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018. 6, 7

[44] 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. 1, 2, 3, 5, 6, 7

[45] 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. 1, 2, 3, 6, 7

[46] Chenshen Wu, Luis Herranz, Xialei Liu, Joost Van De Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. *Advances in Neural Information Processing Systems*, 31, 2018. 3

[47] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021. 3

[48] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 3