Training Consistent Mixture-of-Experts-Based Prompt Generator for Continual Learning

Yue Lu[1], Shizhou Zhang[1*], De Cheng[2*], Guoqiang Liang[1], Yinghui Xing[1], Nannan Wang[2], Yanning Zhang[1]

[1]School of Computer Science, Northwestern Polytechnical University, China
[2]School of Telecommunications Engineering, Xidian University, China
zgxd@mail.nwpu.edu.cn, szzhang@nwpu.edu.cn, dcheng@xidian.edu.cn, gqliang@nwpu.edu.cn, xyh_7491@nwpu.edu.cn, nnwang@xidian.edu.cn, ynzhang@nwpu.edu.cn

Abstract

Visual prompt tuning-based continual learning (CL) methods have shown promising performance in exemplar-free scenarios, where their key component can be viewed as a prompt generator. Existing approaches generally rely on freezing old prompts, slow updating and task discrimination for prompt generators to preserve stability and minimize forgetting. In contrast, we introduce a novel approach that trains a *consistent prompt generator* to ensure stability during CL. *Consistency* means that for any instance from an old task, its corresponding instance-ware prompt generated by the prompt generator *remains consistent* even as the generator continually updates in a new task. This ensures that the representation of a specific instance remains stable across tasks and thereby prevents forgetting. We employ a mixture of experts (MoE) as the prompt generator, which contains a router and multiple experts. By deriving conditions sufficient to achieve the *consistency* for the *MoE prompt generator*, we demonstrate that: during training in a new task, if the router and experts update in the *directions orthogonal* to the subspaces spanned by old input features and gating vectors, respectively, the *consistency* can be theoretically guaranteed. To implement this *orthogonality*, we project parameter gradients to those orthogonal directions using the orthogonal projection matrices computed via the null space method. Extensive experiments on four class-incremental learning benchmarks validate the effectiveness and superiority of our approach.

Introduction

Continual learning (CL) aims to learn new tasks without forgetting the old ones in the scenario of sequentially arrived data (Ratcliff 1990; McCloskey and Cohen 1989). In recent years, visual prompt tuning-based (Jia et al. 2022) CL methods (Wang et al. 2022b,a) utilizing pre-trained Vision Transformers (Dosovitskiy et al. 2021) have shown promising performance in CL. Even without storing exemplars (*i.e.*, samples from previous tasks), they can outperform conventional CNN-based CL approaches by a large margin.

Despite the variations among existing prompt-based CL methods, their fundamental objective is to *generate stable instance-aware prompts* from a form of prompt generator to reduce *catastrophic forgetting* and *maintain stability*. Mainstream methods employ a prompt pool (Wang et al. 2022b), which retrieves an instance-aware prompt based on the input feature, as the prompt generator. They usually keep stability by *freezing old prompts* and incrementally training new task-specific prompts (Wang et al. 2022a; Smith et al. 2023), which is essentially a progressive expansion scheme (Wang et al. 2024). Other methods train prompt generators by *slow updating*, *first-task adaptation* or *task discrimination* for stability (Gao et al. 2023; Jung et al. 2023; Khan et al. 2023). Nevertheless, existing methods cannot theoretically guarantee the use of consistent instance-aware prompts or experts across tasks, leading to representational drift and forgetting.

Different from existing works, we propose to *train a consistent prompt generator to preserve the model's stability* during CL. Specifically, the consistency is that for any instance from the old task, its corresponding instance-aware prompt generated by the prompt generator remains *consistent* as the prompt generator updates continually in a new task. By doing this, the representation of the instance does not drift, and stability can be maintained in theory. The *consistency* can be achieved by *orthogonal projection* constraints on the gradients (Wang et al. 2021; Saha, Garg, and Roy 2021; Qiao et al. 2024; Lu et al. 2024) of the prompt generator during training in the new task. Consequently, our prompt generator has the ability to generate *consistent instance-aware prompts across tasks* to mitigate forgetting.

To more effectively select and integrate diverse knowledge for prompt generation, we use a mixture of experts (MoE) as the prompt generator. The MoE consists of a routing strategy and multiple experts, each contributing specialized knowledge. The routing strategy uses a router to guide the selection of specific experts for each input feature. A fixed set of candidate prompts serves as experts, which are selectively combined to generate the instance-aware prompt. The MoE not only enables each expert to specialize in clusters of similar tasks, but also mitigates forgetting through the correct selection of instance-relevant experts. Therefore, we propose a *consistent MoE prompt generator* to maintain stability for continual learning in our approach.

To derive the specific *orthogonal projections* for the proposed MoE prompt generator, we analyze the conditions that satisfy the *consistency objective*: in the new task where the

*Shizhou Zhang and De Cheng are co-corresponding authors.

router and experts have been updated, the generated prompt for an old input feature should be identical to that generated in the old task. By decoupling this consistency objective into two joint equations related to the router and experts and solving them separately, we derive two sufficient *orthogonality conditions*. They reveal that if 1) the *router* and 2) the *experts* update in the directions orthogonal to the subspaces spanned by 1) *old input features* and 2) *old gating vectors*, *respectively*, the consistency objective can be achieved. To implement the orthogonality conditions, we employ the null space method (Wang et al. 2021) to compute two *orthogonal projection matrices*. To better trade off plasticity against stability, we aim to enable experts to acquire more new knowledge without altering the experts already trained to specialize in handling specific instances. Therefore, we relax the orthogonal constraint for the experts by reducing the weight of orthogonal projection matrix. Then the obtained matrices are used to project gradients during training in the new task.

We summarize our contributions as follows:

- We propose to *train a consistent prompt generator to maintain the model's stability theoretically*, which is distinct from existing prompt generators in the scheme of anti-forgetting.
- We propose *a MoE as the prompt generator*, which can leverage the strengths of individual experts to handle diverse knowledge and adapt to various downstream tasks.
- We derive *two orthogonality conditions* that provide a theoretical guarantee for training the consistent MoE prompt generator. By implementing these conditions, our approach shows solid effectiveness in anti-forgetting and achieves state-of-the-art performance.

Related Work

Prompt-Based Continual Learning The main line of methods designs a prompt pool as the prompt generator to retrieve an instance-aware prompt by key-value matching (Wang et al. 2022b, 2023b; Kurniawan et al. 2024; Xing et al. 2023). They freeze the prompts trained in old tasks to keep stability, such as CODA-Prompt (Smith et al. 2023) and ConvPrompt (Roy et al. 2024). Another line of methods updates parameters at a slow rate or only in the first task to keep the parameters stable (Gao et al. 2023; Jung et al. 2023). For example, LAE (Gao et al. 2023) updates an offline expert with a large momentum to reduce the alteration of parameters. Some works aim to improve stability by enhancing the task discrimination ability. The techniques mainly include task-identity inference network (Wang et al. 2023a; Yu et al. 2024), multi-centroid prototype (Wang, Huang, and Hong 2022; Yang et al. 2023; Li et al. 2024b) and language guidance (Wang et al. 2023b; Khan et al. 2023). For example, S-Prompts (Wang, Huang, and Hong 2022) clusters the class prototypes into multiple centroids for a better matching of task-specific prompts.

MoE in Continual Learning MoEs have emerged as a popular technique for scaling large models efficiently. Some works introduce MoEs (Jacobs et al. 1991; Shazeer et al. 2017) in CL (Yu et al. 2024; Rypeść et al. 2024). ExpertGate (Aljundi, Chakravarty, and Tuytelaars 2017) and DDAS (Yu et al. 2024) incrementally expand or freeze experts with the increase of tasks, and use out-of-distribution detectors implemented by task-specific auto-encoders to determine which expert to route at test time. DSE (Chen et al. 2023) preserves old knowledge by freezing old experts and fits new data distributions by adding experts with regularization. By contrast, our MoE generates consistent prompts across tasks by orthogonal projections, which differs from them in the scheme of maintaining stability.

Orthogonal Projection in Continual Learning A line of CNN-based CL methods focuses on explicitly manipulating the optimization program for anti-forgetting (Saha, Garg, and Roy 2021; Deng et al. 2021; Kong et al. 2022; Lin et al. 2022; Hu et al. 2024). OWM (Zeng et al. 2019) constructs an orthogonal projector to project the gradients of convolutional kernels into the direction orthogonal to the subspace of the old input features in previous tasks. By doing this, the output features for old instances can remain unchanged during training in a new task, which theoretically keeps stability. NSCL (Wang et al. 2021) directly projects the gradients into the null space of old input features, hence becoming more straightforward. We adopt the null space method to compute orthogonal projection matrices in our approach.

However, since the operations in the MoE prompt generator which includes routing and experts are more complicated than convolutional and linear operations, the orthogonality condition for them is inapplicable to our method. We need to analyze the consistency objective specifically for our model and derive concrete orthogonality conditions to implement orthogonal projections, as introduced in the next section.

Method

Problem Setting

Continual learning can be defined as training a model over a sequence of T tasks $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_T$. In the t-th task $\mathcal{T}_t$, the training set is $\mathfrak{D}_t = \{\mathcal{X}_n^t, y_n^t\}_{n=1}^{|\mathcal{T}_t|}$, where $\mathcal{X}_n^t$ is the n-th image with label y_n^t, and $|\mathcal{T}_t|$ denotes the number of samples in this task. We focus on the class-incremental learning protocol: the label space $\mathcal{Y}_t$ of task $\mathcal{T}_t$ is disjoint with other tasks, *i.e.*, $\bigcap_{t=1}^{T} \mathcal{Y}_t = \emptyset$. Once the model finishes learning on task $\mathcal{T}_t$, the corresponding training set $\mathfrak{D}_t$ will be dropped and become inaccessible when learning from $\mathcal{T}_{t+1}$. The model should be able to classify test samples from any learned task.

Overview of Our Approach

We aim to train a consistent prompt generator $\Phi(\boldsymbol{x}; \boldsymbol{\Theta})$ for CL, where $\boldsymbol{x}$ and $\boldsymbol{\Theta}$ denote its input feature and trainable parameters, respectively. As illustrated in Figure 1, for an input feature $\boldsymbol{x}^t$ extracted from an instance $\mathcal{X}^t$ (the instance index is omitted) which belongs to the old task $\mathcal{T}_t$, we aim to achieve the following *consistency objective* formulated as:

$$\Phi(\boldsymbol{x}^t; \boldsymbol{\Theta}^t) = \Phi(\boldsymbol{x}^t; \boldsymbol{\Theta}^{t+1}), \tag{1}$$

where $\boldsymbol{\Theta}^t$ and $\boldsymbol{\Theta}^{t+1}$ denote the parameters after training on the old task $\mathcal{T}_t$ and the new task $\mathcal{T}_{t+1}$, respectively. We use $\Delta\boldsymbol{\Theta}$ to denote the parameter update in the new task, *i.e.*, $\boldsymbol{\Theta}^{t+1} = \boldsymbol{\Theta}^t + \Delta\boldsymbol{\Theta}$. $\Delta\boldsymbol{\Theta}$ is the variable to be solved for

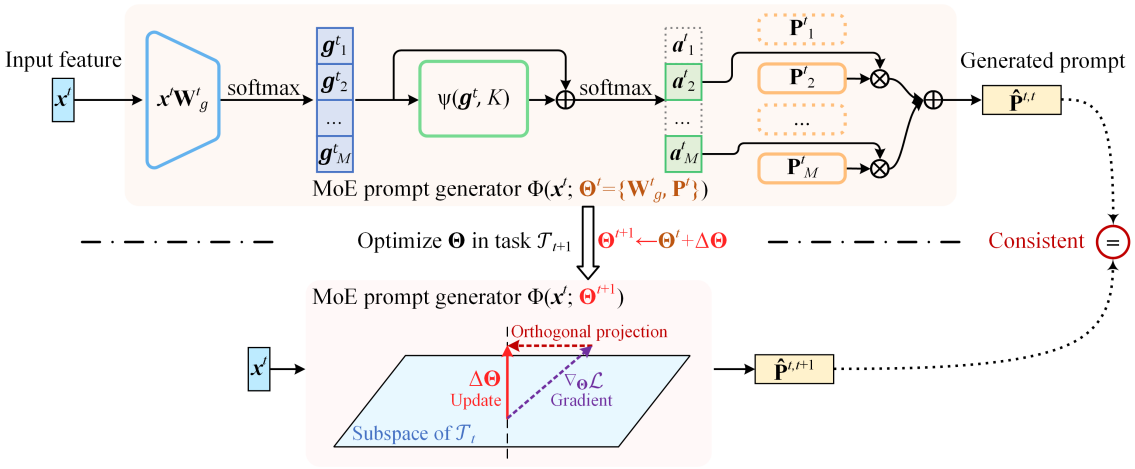

Figure 1: Illustration of our approach. $\mathbf{W}_g^t$: the weight matrix of the router which outputs M gating values; $\mathbf{P}^t$: the concatenation of experts; superscript of a variable: the task identifier; subscript in $\boldsymbol{g}^t$, $\boldsymbol{a}^t$ and $\mathbf{P}^t$: the component index in the vector/matrix; $\Psi(\cdot)$: the function defined for selecting top-K gates; $\otimes$: product; $\oplus$: addition. The dashed blocks denote the unactivated gates or experts. Our objective is to generate consistent prompts (*i.e.*, $\hat{\mathbf{P}}^{t,t+1} = \hat{\mathbf{P}}^{t,t}$) for the old feature ($\boldsymbol{x}^t$) of the old task ($\mathcal{T}_t$) when the parameters ($\boldsymbol{\Theta}^t$) of the MoE prompt generator ($\Phi(\cdot)$) update (by $\Delta\boldsymbol{\Theta}$) in the new task ($\mathcal{T}_{t+1}$).

Eq. (1). Specifically, in the case of our prompt generator implemented by MoE, $\Delta\boldsymbol{\Theta} = \{\Delta\mathbf{W}_g, \Delta\mathbf{P}\}$, where $\mathbf{W}_g$ is the weight matrix of the router and $\mathbf{P}$ denotes the candidate prompts served as experts.

A Vision Transformer (ViT) composed of several ViT layers is used as the backbone. Suppose a ViT layer $\Omega(\cdot)$ where its layer index is omitted. We use $\mathbf{E}$ and $\boldsymbol{x}$ to denote the image tokens and class token to be fed into $\Omega(\cdot)$, respectively. $\boldsymbol{x}$ is also the input of the prompt generator $\Phi(\cdot)$ which generates the prompt: $\hat{\mathbf{P}} = \Phi(\boldsymbol{x}; \boldsymbol{\Theta})$. Then $\hat{\mathbf{P}}$ is fed into $\Omega(\cdot)$ with $\mathbf{E}$ and $\boldsymbol{x}$ to obtain the output of the ViT layer: $\mathbf{Y} = \Omega([\boldsymbol{x}, \hat{\mathbf{P}}, \mathbf{E}])$. The output class token of the last ViT layer will be fed into a classifier for classification. Note that the prompt generator is applicable to any ViT layer that needs prompt. Overall, our *consistent MoE prompt generator* mainly contains the MoE and consistent prompt generation, which are introduced below.

MoE Prompt Generator

The detailed structure of our MoE prompt generator $\Phi(\cdot)$ is shown in Figure 1, which consists of routing and experts.

Routing The input feature (*i.e.*, the class token) $\boldsymbol{x} \in \mathbb{R}^D$ first undergoes the router with a $\mathrm{softmax}$ operation to obtain the gates $\boldsymbol{g}$, with D denoting the dimension of the feature. Suppose there are M experts in the MoE, and thereby $\boldsymbol{g}$ is a vector containing M gates:

$$\boldsymbol{g} = \mathrm{softmax}(\boldsymbol{x}\mathbf{W}_g) \in \mathbb{R}^M, \tag{2}$$

where $\mathbf{W}_g \in \mathbb{R}^{D\times M}$ is the weight matrix of the router.

In a typical MoE, only K maximum gates are selected, and the corresponding K experts will be activated afterwards. To denote the sparse operation of selecting top-K gates in a dense manner, we define a function $\Psi(\cdot)$ to get a mask according to the gates in $\boldsymbol{g}$:

$$\Psi(\boldsymbol{g}, K) = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_M], \tag{3}$$

where $\boldsymbol{d}_m$ denotes the m-th value in $\boldsymbol{d}$. $\forall m \in \{1, \cdots, M\}$,

$$\boldsymbol{d}_m = \begin{cases} 0, & \text{if } \boldsymbol{g}_m \text{ is in the top } K \text{ elements of } \boldsymbol{g}, \\ -\infty, & \text{otherwise.} \end{cases} \tag{4}$$

We can use the function $\Psi(\cdot)$ to mask the non-top-K gates in $\boldsymbol{g}$ through adding $\Psi(\boldsymbol{g}, K)$ to $\boldsymbol{g}$. As a result, the masked gates are set to $-\infty$, while the selected top-K gates are retained. Then we normalize the top-K gates by $\mathrm{softmax}$ so that their sum equals 1, while the unactivated gates are set to 0. By doing this, we obtain the gating vector $\boldsymbol{a}$:

$$\boldsymbol{a} = \mathrm{softmax}(\boldsymbol{g} + \Psi(\boldsymbol{g}, K)) \in \mathbb{R}^M. \tag{5}$$

Experts The concatenation of all the experts is denoted as $\mathbf{P} = [\mathbf{P}_1, \cdots, \mathbf{P}_M] \in \mathbb{R}^{M\times L\times D}$, where L is the prompt length. The expert whose corresponding gating value in $\boldsymbol{a}$ is greater than 0 will be activated. We fuse the activated experts by a weighted sum to obtain the generated prompt $\hat{\mathbf{P}}$. This sparse fusion can be represented in a dense form as:

$$\hat{\mathbf{P}} = \boldsymbol{a}\mathbf{P} \in \mathbb{R}^{L\times D}, \tag{6}$$

We further analyze how to *generate consistent prompts* from the proposed MoE prompt generator as follows.

Conditions for Consistent Prompt Generation

As aforementioned, $\boldsymbol{\Theta}$ is composed of $\mathbf{W}_g$ and $\mathbf{P}$ in our MoE prompt generator. Therefore, our *consistency objective* of Eq. (1) can be reformulated as:

$$\Phi(\boldsymbol{x}^t; \mathbf{W}_g^t, \mathbf{P}^t) = \Phi(\boldsymbol{x}^t; \mathbf{W}_g^{t+1}, \mathbf{P}^{t+1}), \tag{7}$$

where $\mathbf{W}_g^{t+1} = \mathbf{W}_g^t + \Delta\mathbf{W}_g$ and $\mathbf{P}^{t+1} = \mathbf{P}^t + \Delta\mathbf{P}$. $\Delta\mathbf{W}_g$ and $\Delta\mathbf{P}$ represent the parameter updates in task $\mathcal{T}_{t+1}$.

We aim to derive *the conditions expressed with* $\Delta\mathbf{W}_g$ *and* $\Delta\mathbf{P}$ by solving Eq. (7), such that we can perform projections on parameter gradients to achieve this consistency objective for stability. Nevertheless, it is difficult to derive the necessary and sufficient conditions for Eq. (7). The reason is that there are *non-injective* functions (*e.g.*, Eq. (4) and $\mathrm{softmax}$) which result in non-unique (infinite) solutions. We turn to derive one (or more) *sufficient condition(s)* that can satisfy Eq. (7), since a sufficient condition can be seen as a *particular solution* to the equation.

To analyze the consistency objective, we formulate Eq. (7) as the following equation according to Eq. (6):

$$\boldsymbol{a}^t\mathbf{P}^t = \boldsymbol{a}^{t+1}\mathbf{P}^{t+1}. \tag{8}$$

Since there are two potential independent variables (*i.e.*, $\Delta\mathbf{W}_g$ and $\Delta\mathbf{P}$) in this only equation, we adopt a two-step strategy to solve it. Firstly, we assume the following equation holds:

$$\boldsymbol{a}^t = \boldsymbol{a}^{t+1}. \tag{9}$$

Then Eq. (8) can be simplified as the following equation based on Eq. (9):

$$\boldsymbol{a}^t\mathbf{P}^t = \boldsymbol{a}^t\mathbf{P}^{t+1}. \tag{10}$$

In this way, we can solve Eq. (8) by solving Eq. (9) and Eq. (10) jointly.

We first analyze the sufficient condition for satisfying Eq. (9) based on Eq. (5):

$$\mathrm{softmax}(\boldsymbol{g}^t + \Psi(\boldsymbol{g}^t, K)) = \mathrm{softmax}(\boldsymbol{g}^{t+1} + \Psi(\boldsymbol{g}^{t+1}, K)) \tag{11}$$

It is difficult to solve the above equation due to the aforementioned non-injection property. However, it is clear to see that if $\boldsymbol{g}^t$ equals $\boldsymbol{g}^{t+1}$, then $\Psi(\boldsymbol{g}^t, K) = \Psi(\boldsymbol{g}^{t+1}, K)$ holds, and hence Eq. (11) is also true. Therefore, we derive the following sufficient condition that satisfies Eq. (11):

$$\boldsymbol{g}^t = \boldsymbol{g}^{t+1}, \tag{12}$$

which is also the sufficient condition satisfying Eq. (9). According to Eq. (2), Eq. (12) can be expanded as:

$$\mathrm{softmax}(\boldsymbol{x}^t\mathbf{W}_g^t) = \mathrm{softmax}(\boldsymbol{x}^t\mathbf{W}_g^{t+1}). \tag{13}$$

A sufficient condition for Eq. (13) is:

$$\boldsymbol{x}^t\mathbf{W}_g^t = \boldsymbol{x}^t\mathbf{W}_g^{t+1} = \boldsymbol{x}^t(\mathbf{W}_g^t + \Delta\mathbf{W}_g). \tag{14}$$

Consequently, we derive the following equation expressed with $\Delta\mathbf{W}_g$, which is the sufficient condition for Eq. (9):

$$\boldsymbol{x}^t\Delta\mathbf{W}_g = \mathbf{0}. \tag{15}$$

Next, we analyze the sufficient condition for satisfying Eq. (10) by substituting $\mathbf{P}^{t+1} = \mathbf{P}^t + \Delta\mathbf{P}$:

$$\boldsymbol{a}^t\mathbf{P}^t = \boldsymbol{a}^t\mathbf{P}^{t+1} = \boldsymbol{a}^t(\mathbf{P}^t + \Delta\mathbf{P}), \tag{16}$$

which is further simplified as:

$$\boldsymbol{a}^t\Delta\mathbf{P} = \mathbf{0}. \tag{17}$$

As deduced above, we finally derive two sufficient conditions that jointly achieve the consistency objective of Eq. (7):

$$\begin{cases} \boldsymbol{x}^t\Delta\mathbf{W}_g = \mathbf{0} & (18) \\ \boldsymbol{a}^t\Delta\mathbf{P} = \mathbf{0} & (19) \end{cases}$$

Eq. (18) and (19) indicate that during training the MoE prompt generator in task $\mathcal{T}_{t+1}$, if the router update ($\Delta\mathbf{W}_g$) and the experts update ($\Delta\mathbf{P}$) can be *orthogonal to the subspaces spanned by the old input feature* $\boldsymbol{x}^t$ *and gating vector* $\boldsymbol{a}^t$, respectively, the generated prompt for that old input feature will keep consistent. We refer to these two equations as the *orthogonality conditions* for our consistency objective.

Computation of Orthogonal Projection Matrices

The gradients of $\mathbf{W}_g$ and $\mathbf{P}$ calculated in task $\mathcal{T}_{t+1}$ are denoted as $\nabla_{\mathbf{W}_g}\mathcal{L}$ and $\nabla_{\mathbf{P}}\mathcal{L}$, respectively, where $\mathcal{L}$ is the loss. We aim to *compute two projection matrices* $\mathbf{H}_w$ *and* $\mathbf{H}_p$ which can project the gradients as the parameter updates to satisfy the orthogonality conditions:

$$\begin{cases} \Delta\mathbf{W}_g = \mathbf{H}_w\nabla_{\mathbf{W}_g}\mathcal{L}, & (20) \\ \Delta\mathbf{P} = \mathbf{H}_p\nabla_{\mathbf{P}}\mathcal{L}, & (21) \end{cases}$$

The null space method (Wang et al. 2021) is adopted in our approach to obtain the projection matrices as detailed below.

We use the superscript t, n to denote the index of the n-th instance in task $\mathcal{T}_t$ ($n \in \{1, \cdots, |\mathcal{T}_t|\}$). The concatenation matrices of $\boldsymbol{x}^t$ and $\boldsymbol{a}^t$ for all the instances in $\mathcal{T}_t$ are represented as $\mathbf{X}^t = \left[\boldsymbol{x}^{t,1}, \cdots, \boldsymbol{x}^{t,|\mathcal{T}_t|}\right] \in \mathbb{R}^{|\mathcal{T}_t|\times D}$ and $\mathbf{A}^t = \left[\boldsymbol{a}^{t,1}, \cdots, \boldsymbol{a}^{t,|\mathcal{T}_t|}\right] \in \mathbb{R}^{|\mathcal{T}_t|\times M}$, respectively. First, we compute the uncentered covariance matrices for the two concatenation matrices: $\bar{\mathbf{X}}^t = (\mathbf{X}^t)^\top\mathbf{X}^t$ and $\bar{\mathbf{A}}^t = (\mathbf{A}^t)^\top\mathbf{A}^t$.

Then we perform singular value decomposition (SVD) on the uncentered covariance matrices:

$$\mathbf{U}_x\mathbf{\Lambda}_x\mathbf{V}_x^\top = \bar{\mathbf{X}}^t, \quad \mathbf{U}_a\mathbf{\Lambda}_a\mathbf{V}_a^\top = \bar{\mathbf{A}}^t. \tag{22}$$

Next, a total of R_x right singular vectors whose corresponding singular values close to zero are picked from $\mathbf{V}_x$. The matrix composed of those R_x vectors is denoted as $\tilde{\mathbf{V}}_x \in \mathbb{R}^{D\times R_x}$, which represents the bases of the (approximate) null space of $\bar{\mathbf{X}}^t$. We follow (Wang et al. 2021) to regard the singular values lower than $\alpha_x\lambda_{x,\min}$ to be close to zero so as to determine R_x. α_x is a hyper-parameter and $\lambda_{x,\min}$ denotes the minimum non-zero singular value in $\mathbf{\Lambda}_x$. Similarly, we can obtain a matrix $\tilde{\mathbf{V}}_a \in \mathbb{R}^{M\times R_a}$ representing the bases of the null space of $\bar{\mathbf{A}}^t$, where R_a denotes its nullity determined by a hyper-parameter α_a.

Finally, the projection matrices $\mathbf{H}_w$ and $\mathbf{H}_p$ are derived:

$$\mathbf{H}_w = \tilde{\mathbf{V}}_x\tilde{\mathbf{V}}_x^\top, \quad \mathbf{H}_p = \tilde{\mathbf{V}}_a\tilde{\mathbf{V}}_a^\top. \tag{23}$$

To sum up, we use Eq. (20) and (21) to perform orthogonal projections, where the projection matrices are computed by Eq. (23). By doing this, we achieve the objective of *consistent prompt generation* for the old input features in task $\mathcal{T}_t$ during training in task $\mathcal{T}_{t+1}$. To further balance stability and plasticity better, we assign a weight $\eta \in [0, 1]$ to the projection matrix of experts ($\mathbf{H}_p$) to relax the orthogonal constraint and enhance plasticity: $\Delta\mathbf{P} = [\eta\mathbf{H}_p + (1-\eta)\mathbf{I}]\nabla_{\mathbf{P}}\mathcal{L}$, where η should be close to 1.

As for all previously learned tasks $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_t$, we perform SVD on the accumulated uncentered covariance matrices $\sum_{\tau=1}^t \bar{\mathbf{X}}^\tau$ and $\sum_{\tau=1}^t \bar{\mathbf{A}}^\tau$, and then calculate $\mathbf{H}_w$ and $\mathbf{H}_p$ according to Eq. (23) in the same manner. Consequently, the generated prompts can retain consistent for the instances across all the learned tasks. An algorithm of our approach is provided in the Algorithm section of the Appendix.

Method	10-split CIFAR-100		20-split CIFAR-100		10-split ImageNet-R		10-split DomainNet	
	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓
VPT-baseline	$84.68_{\pm 0.23}$	$15.36_{\pm 0.34}$	$80.78_{\pm 0.34}$	$19.11_{\pm 0.35}$	$72.37_{\pm 0.24}$	$19.16_{\pm 0.29}$	$73.31_{\pm 0.28}$	$27.18_{\pm 0.36}$
VPT-CPG	$\mathbf{90.63}_{\pm 0.44}$	$\mathbf{3.98}_{\pm 0.65}$	$\mathbf{88.08}_{\pm 0.77}$	$\mathbf{5.20}_{\pm 0.64}$	$\mathbf{78.63}_{\pm 0.52}$	$\mathbf{7.18}_{\pm 0.62}$	$\mathbf{83.21}_{\pm 0.67}$	$\mathbf{7.09}_{\pm 0.82}$
Upper-bound	$93.57_{\pm 0.07}$	-	$93.57_{\pm 0.07}$	-	$84.91_{\pm 0.17}$	-	$89.42_{\pm 0.04}$	-
CLIP-baseline	$75.40_{\pm 0.73}$	$19.18_{\pm 0.98}$	$71.97_{\pm 1.41}$	$21.44_{\pm 1.74}$	$79.14_{\pm 0.29}$	$9.62_{\pm 0.79}$	$84.65_{\pm 0.19}$	$10.31_{\pm 0.33}$
CLIP-CPG	$\mathbf{82.76}_{\pm 0.44}$	$\mathbf{6.14}_{\pm 0.56}$	$\mathbf{82.06}_{\pm 0.77}$	$\mathbf{5.68}_{\pm 0.87}$	$\mathbf{82.08}_{\pm 0.55}$	$\mathbf{6.28}_{\pm 0.63}$	$\mathbf{88.78}_{\pm 0.30}$	$\mathbf{4.11}_{\pm 0.50}$
Upper-bound	$86.44_{\pm 0.21}$	-	$86.44_{\pm 0.21}$	-	$85.05_{\pm 0.03}$	-	$91.07_{\pm 0.04}$	-

Table 1: Comparison between the proposed approach ("-CPG") and the baseline of sequential fine-tuning using VPT and CLIP models. The upper-bound means jointly training all the classes in the dataset. The value after ± indicates the standard deviation.

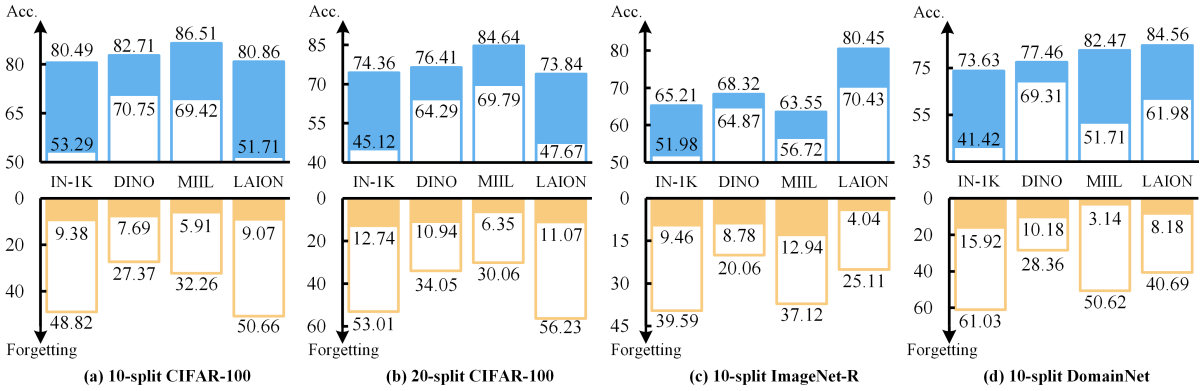

Figure 2: Results of using different types of pre-training parameters for the VPT-based model. The annotated values of the filled bars denote the accuracy or forgetting of our approach, while those of the blank bars denote the two metrics of the baseline.

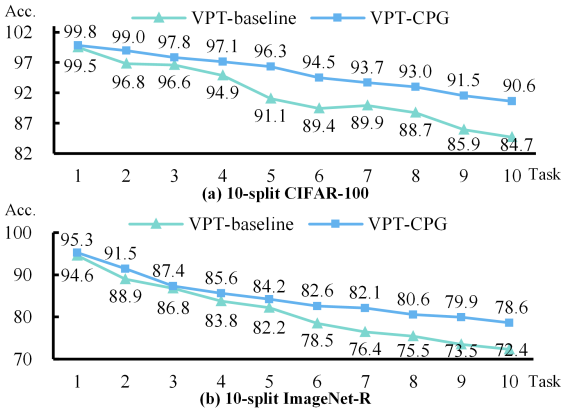

Figure 3: Task-by-task accuracy changing curves of the VPT-baseline and the VPT-CPG on two benchmarks.

Experiments

Experimental Setups

We evaluate our approach using four class-incremental learning (CIL) benchmarks: 10-split and 20-split CIAFR-100 (Krizhevsky and Hinton 2009), 10-split ImageNet-R (Hendrycks et al. 2021) and 10-split DomainNet (Peng et al. 2019). Note that the 10-split DomainNet is organized by (Wang et al. 2023c) specifically for cross-domain CIL, with the top 200 classes from the original DomainNet (Peng et al. 2019) selected based on the number of images.

We employ the ViT-B/16 (Dosovitskiy et al. 2021) as the backbone for all experiments. Each of the 12 ViT layers is equipped with our proposed prompt generator by default. The MoE includes 36 experts ($M = 36$), and we select the top four experts ($K = 4$) for CIFAR-100 and ImageNet-R, while 16 experts are chosen for DomainNet. To enhance the training of MoE, we implement the balancing-expert strategy as proposed by (Shazeer et al. 2017). We report the mean values of the final average accuracy and final average forgetting over three runs with different random seeds. Additional experimental details can be found in the Appendix. Our code is available at https://github.com/zugexiaodui/ConsistentMoEPromptGenerator.

Experimental Results

Validation of Effectiveness To comprehensively validate the effectiveness of our approach, we conduct experiments on four benchmarks using two types of models involving a total of six types of pre-training parameters. The baseline, which employs sequential fine-tuning without the MoE and orthogonal projections, *i.e.*, only one expert without routing, is represented by "-baseline". Our consistent MoE prompt generator is denoted by "-CPG". The models include the ViT pre-trained on ImageNet-21K (Russakovsky et al. 2015),

Method	Venue	10-split CIFAR-100		20-split CIFAR-100		10-split ImageNet-R		10-split DomainNet	
		Acc.↑	Forgetting↓	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓
L2P	CVPR'22	$83.83_{\pm 0.04}$	$7.63_{\pm 0.30}$	$80.10_{\pm 0.72}^{\ddagger}$	-	$61.57_{\pm 0.66}$	$9.73_{\pm 0.47}$	$81.17_{\pm 0.83}^{\dagger}$	$8.98_{\pm 1.25}$
DualPrompt	ECCV'22	$86.51_{\pm 0.33}$	$5.16_{\pm 0.09}$	$82.02_{\pm 0.32}^{\ddagger}$	-	$68.13_{\pm 0.49}$	$4.68_{\pm 0.20}$	$81.70_{\pm 0.78}^{\dagger}$	$8.04_{\pm 0.31}$
CODA-Prompt	CVPR'23	$86.25_{\pm 0.74}$	$1.67_{\pm 0.26}$	-	-	$75.45_{\pm 0.56}$	$1.64_{\pm 0.10}$	$80.04_{\pm 0.79}^{\dagger}$	$10.16_{\pm 0.35}$
LAE	ICCV'23	$85.59_{\pm 0.46}$	-	$83.93_{\pm 0.28}$	-	$72.66_{\pm 0.63}$	-	-	-
LGCL	ICCV'23	$87.23_{\pm 0.21}$	$5.10_{\pm 0.15}$	-	-	$69.46_{\pm 0.04}$	$4.20_{\pm 0.06}$	-	-
C-LN	ICCVW'23	$86.95_{\pm 0.37}$	$6.98_{\pm 0.43}$	-	-	$76.36_{\pm 0.51}$	$8.31_{\pm 1.28}$	-	-
ESN	AAAI'23	$86.34_{\pm 0.52}$	$4.76_{\pm 0.14}$	$80.56_{\pm 0.94}^{\ddagger}$	-	$62.61_{\pm 0.96}^{\ddagger}$	-	$79.22_{\pm 2.04}^{\dagger}$	$10.62_{\pm 2.12}$
HSICBO	AAAI'24	-	-	-	-	$71.43_{\pm 0.22}$	-	-	-
EvoPrompt	AAAI'24	$87.97_{\pm 0.30}$	$2.60_{\pm 0.42}$	$84.64_{\pm 0.14}$	$3.98_{\pm 0.24}$	$76.83_{\pm 0.08}$	$2.78_{\pm 0.06}$	$79.50_{\pm 0.29}^{*}$	$3.81_{\pm 0.36}$
PGP	ICLR'24	$86.92_{\pm 0.05}$	$5.35_{\pm 0.19}$	$83.74_{\pm 0.01}$	$7.91_{\pm 0.15}$	$69.34_{\pm 0.05}$	$4.53_{\pm 0.04}$	$80.41_{\pm 0.25}^{*}$	$8.39_{\pm 0.18}$
OVOR-Deep	ICLR'24	$85.99_{\pm 0.89}$	$6.42_{\pm 2.03}$	$84.13_{\pm 0.75}^{*}$	$6.81_{\pm 0.77}$	$76.11_{\pm 0.21}$	$7.16_{\pm 0.34}$	$79.61_{\pm 0.86}^{*}$	$4.77_{\pm 0.94}$
ConvPrompt	CVPR'24	$\underline{88.87}_{\pm 0.33}$	$4.75_{\pm 0.15}$	$\underline{87.22}_{\pm 0.42}^{*}$	$5.43_{\pm 0.29}$	$\underline{77.86}_{\pm 0.25}$	$4.33_{\pm 0.24}$	$79.47_{\pm 0.35}^{*}$	$6.49_{\pm 0.43}$
InfLoRA	CVPR'24	$87.06_{\pm 0.25}$	$6.22_{\pm 0.39}$	$81.42_{\pm 0.54}^{*}$	$6.42_{\pm 0.33}$	$75.65_{\pm 0.14}$	$5.73_{\pm 0.44}$	$81.45_{\pm 0.68}^{*}$	$5.35_{\pm 0.52}$
EASE	CVPR'24	87.76	5.94	85.80	7.19	76.17	7.82	78.89*	7.89
CPrompt	CVPR'24	$87.82_{\pm 0.21}$	$5.06_{\pm 0.50}$	$83.97_{\pm 0.31}^{*}$	$6.85_{\pm 0.43}$	$77.14_{\pm 0.11}$	$5.97_{\pm 0.68}$	$\underline{82.97}_{\pm 0.34}$	$7.45_{\pm 0.93}$
VPT-CPG	This work	$\mathbf{90.63}_{\pm 0.44}$	$3.98_{\pm 0.65}$	$\mathbf{88.08}_{\pm 0.77}$	$5.20_{\pm 0.64}$	$\mathbf{78.63}_{\pm 0.52}$	$7.18_{\pm 0.62}$	$\mathbf{83.21}_{\pm 0.67}$	$7.09_{\pm 0.82}$

Table 2: Comparison with existing methods whose backbones are pre-trained on the ImageNet-21K. The results marked with †, ‡ and * are implemented by (Gao, Cen, and Chang 2024), (Gao et al. 2023) and us, respectively, due to lack of official results. The highest accuracies are in **bold**, and the second highest accuracies are underlined.

MoE	$\mathbf{H}_w$	$\mathbf{H}_p$	10-split CIFAR-100		20-split CIFAR-100		10-split ImageNet-R		10-split DomainNet	
			Acc.↑	Forgetting↓	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓
×	×	×	$84.68_{\pm 0.23}$	$15.36_{\pm 0.34}$	$80.78_{\pm 0.34}$	$19.11_{\pm 0.35}$	$72.37_{\pm 0.24}$	$19.16_{\pm 0.29}$	$73.31_{\pm 0.28}$	$27.18_{\pm 0.36}$
√	×	×	$86.08_{\pm 0.52}$	$13.33_{\pm 0.56}$	$82.06_{\pm 0.75}$	$18.77_{\pm 0.71}$	$74.02_{\pm 0.51}$	$18.33_{\pm 0.69}$	$75.73_{\pm 0.65}$	$22.85_{\pm 0.74}$
√	√	×	$85.72_{\pm 0.41}$	$13.93_{\pm 0.70}$	$80.35_{\pm 0.68}$	$20.54_{\pm 0.79}$	$71.07_{\pm 0.64}$	$20.83_{\pm 0.54}$	$75.26_{\pm 0.56}$	$24.18_{\pm 0.64}$
√	×	√	$88.17_{\pm 0.49}$	$9.89_{\pm 0.60}$	$86.04_{\pm 0.82}$	$12.29_{\pm 0.61}$	$75.83_{\pm 0.63}$	$13.75_{\pm 0.64}$	$80.82_{\pm 0.62}$	$15.68_{\pm 0.84}$
√	√	√	$\mathbf{90.63}_{\pm 0.44}$	$\mathbf{3.98}_{\pm 0.65}$	$\mathbf{88.08}_{\pm 0.77}$	$\mathbf{5.20}_{\pm 0.64}$	$\mathbf{78.63}_{\pm 0.52}$	$\mathbf{7.18}_{\pm 0.62}$	$\mathbf{83.21}_{\pm 0.67}$	$\mathbf{7.09}_{\pm 0.82}$

Table 3: Ablation studies of the three components proposed in our approach. The prompt generator consisting of only one expert without a router is implemented as the baseline of no MoE.

referred to as "VPT" (Jia et al. 2022), and the CLIP pre-trained on WebImageText (Radford et al. 2021), referred to as "CLIP." For the CLIP model, prompt generators are inserted into the image encoder. Table 1 presents the results for both types of models. Our approach consistently outperforms the baseline, enhancing accuracy by 6%∼10% and reducing forgetting by 11%∼20% for the VPT-based model across all benchmarks. Figure 3 visually depicts the accuracy curves for the 10-split CIFAR-100 and ImageNet-R, demonstrating the sustained superiority of our method over the baseline on each task. The CLIP-based model also benefits from our approach, achieving 3%∼10% improvement in accuracy and 3%∼15% reduction in forgetting.

We explore four additional types of pre-training parameters for the VPT-based model, as shown in Figure 2. The pre-training parameters are derived from: naive classification on ImageNet-1K (IN-1K), DINO (Caron et al. 2021) on ImageNet-1K (DINO), MIIL (Ridnik et al. 2021) on ImageNet21k-P (MIIL) and CLIP on LAION-2B (Cherti et al. 2023) (LAION). The significant performance enhancement demonstrates the good generalizability of our method.

Comparison with Existing Approaches We compare the proposed approach with existing state-of-the-art methods on the four benchmarks in Table 2. The competitors include: L2P (Wang et al. 2022b), DualPrompt (Wang et al. 2022a), CODA-Prompt (Smith et al. 2023), LAE (Gao et al. 2023), LGCL (Khan et al. 2023), C-LN (Min et al. 2023), EvoPrompt (Kurniawan et al. 2024), ESN (Wang et al. 2023c), HSICBO (Li et al. 2024a), PGP (Qiao et al. 2024), OVOR-Deep (Huang, Chen, and Hsu 2024), ConvPrompt (Roy et al. 2024), InfLoRA (Liang and Li 2024), EASE (Zhou et al. 2024) and CPrompt (Gao, Cen, and Chang 2024). VPT-CPG surpasses other leading methods by a maximum of 1.76% and an average of 0.91% in accuracy across the four benchmarks, and achieves new state-of-the-art performance.

Ablation Study Our approach comprises three key components: the MoE and two orthogonal projection matrices $\mathbf{H}_w$ and $\mathbf{H}_p$. We study the effect of each component, as shown in Table 3. The proposed MoE improves accuracy

Method	Venue	50S-CIFAR-100		50S-ImageNet-R		50S-DomainNet		100S-ImageNet-R		100S-DomainNet	
		Acc.	Forgetting	Acc.	Forgetting	Acc.	Forgetting	Acc.	Forgetting	Acc.	Forgetting
L2P	CVPR'22	79.90	11.33	51.38	12.34	63.13	11.19	41.51	14.48	54.83	14.95
OVOR-Deep	ICLR'24	67.94	12.79	63.25	5.23	68.29	6.85	43.02	7.30	52.09	8.66
ConvPrompt	CVPR'24	87.55	5.19	64.61	7.12	71.76	6.37	44.32	8.97	56.21	6.27
InfLoRA	CVPR'24	65.47	12.43	62.81	10.37	71.87	9.20	42.23	14.17	48.06	15.43
EASE	CVPR'24	80.28	8.47	70.27	6.73	65.34	9.64	51.56	7.56	37.26	29.12
CPrompt	CVPR'24	78.03	6.57	70.75	7.44	70.74	9.01	59.90	9.52	57.60	9.42
VPT-baseline	Baseline	66.57	33.28	60.22	34.62	46.90	56.12	51.92	43.59	25.03	77.15
VPT-CPG	This work	**87.95**	5.42	**73.08**	12.12	**72.27**	19.93	**64.63**	12.18	**60.81**	27.43

Table 4: Results for long-term continual learning under the settings of 50 tasks and 100 tasks across five benchmarks.

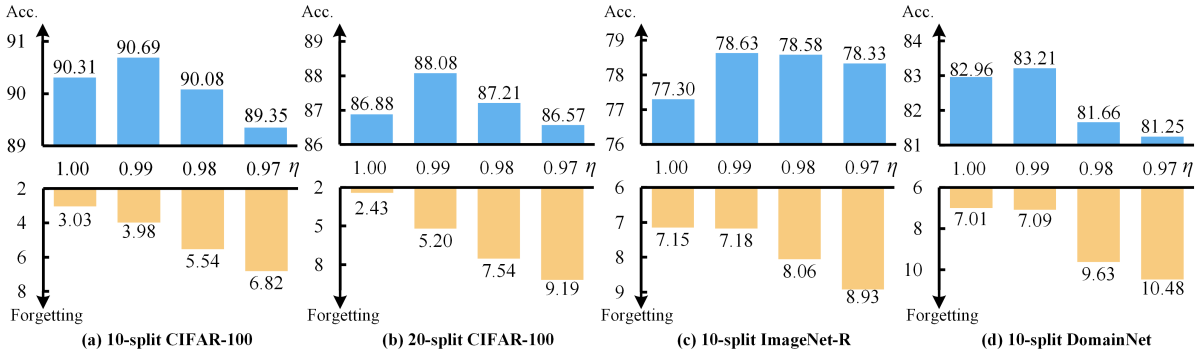

Figure 4: Effects of the orthogonal projection weight η on accuracy and forgetting for the stability-plasticity trade-off.

by an average of 1.69%. As for the orthogonal projections, the projection matrix of the router ($\mathbf{H}_w$) may not solely contribute to reducing forgetting. This is because the same experts are always selected for a specific instance, leading to the complete overwriting of the learned instance-specific knowledge in subsequent tasks. However, when the two projection matrices are used jointly, accuracy is improved and forgetting is mitigated significantly, and the model achieves the best accuracy with the least forgetting. Specifically, the accuracy improves by 4.55%∼7.48% and the forgetting declines by 9.32%∼15.76%. This demonstrates that the two orthogonal projections can collectively contribute to the stability of the MoE prompt generator significantly.

Long-Term Continual Learning In order to verify that our approach remains effective and superior in long-term continual learning, we conduct experiments under the protocols of 50 tasks and 100 tasks across five benchmarks. Besides, we reproduce 6 existing methods for a comparison, as shown in Table 4. It can be seen that VPT-CPG achieves significant improvement in accuracy (by 12%∼35%) and remarkable reduction in forgetting (by 22%∼49%) compared to the VPT-baseline. Compared with other stat-of-the-art approaches, VPT-CPG can still outperform them. It surpasses the second-best approaches by an average of 2.21% with a maximum of 4.7%. This demonstrates that our approach has a good ability to handle long-term CL problem.

Trade-off between Stability and Plasticity Figure 4 illustrates the impacts of η on accuracy and forgetting. When η decreases from 1 to 0.97 in steps of 0.01, the accuracy initially increases and then decreases, while forgetting steadily increases. This indicates that stability weakens while plasticity strengthens. Although forgetting is minimized when η is 1, accuracy is not maximized as the model struggles to learn new knowledge. The model achieves the highest accuracy at the optimal trade-off (η=0.99) between stability and plasticity. This demonstrates the effectiveness of the proposed projection weight in promoting the model to learn new knowledge. More experiments regarding the configurations and analysis of our method can be found in the Appendix.

Conclusion

We propose a *consistent MoE prompt generator* to maintain stability for continual learning in this work. Concretely, for a specific instance, we aim to achieve that the corresponding generated instance-aware prompt in a new task keeps *consistent* with that in the old task, even if the generator updates in the new task. The consistency can be theoretically guaranteed if the router and experts update in the directions *orthogonal* to the subspaces spanned by old input features and gating vectors, respectively. We employ the null space method to compute orthogonal projection matrices to implement this orthogonality. Our approach demonstrates robust effectiveness across various benchmarks in anti-forgetting. As a result, it outperforms existing competitors and achieves state-of-the-art performance.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62101453, 62176198, 62476223, 62201467; in part by Innovation Capability Support Program of Shaanxi (Program No. 2024ZC-KJXX-043); in part by the Young Talent Fund of Xi'an Association for Science and Technology under Grant 959202313088; in part by the Key R&D Program of Shaanxi Province under Grant 2024GX-YBXM135.

References

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T. 2017. Expert Gate: Lifelong Learning with a Network of Experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7120–7129.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9630–9640.

Chen, W.; Zhou, Y.; Du, N.; Huang, Y.; Laudon, J.; Chen, Z.; and Cui, C. 2023. Lifelong Language Pretraining with Distribution-Specialized Experts. In *International Conference on Machine Learning*, volume 202, 5383–5395.

Cherti, M.; Beaumont, R.; Wightman, R.; Wortsman, M.; Ilharco, G.; Gordon, C.; Schuhmann, C.; Schmidt, L.; and Jitsev, J. 2023. Reproducible Scaling Laws for Contrastive Language-Image Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2818–2829.

Deng, D.; Chen, G.; Hao, J.; Wang, Q.; and Heng, P.-A. 2021. Flattening Sharpness for Dynamic Gradient Projection Memory Benefits Continual Learning. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 34, 18710–18721.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Gao, Q.; Zhao, C.; Sun, Y.; Xi, T.; Zhang, G.; Ghanem, B.; and Zhang, J. 2023. A Unified Continual Learning Framework with General Parameter-Efficient Tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11449–11459.

Gao, Z.; Cen, J.; and Chang, X. 2024. Consistent Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 28463–28473.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; Song, D.; Steinhardt, J.; and Gilmer, J. 2021. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8320–8329.

Hu, Y.; Cheng, D.; Zhang, D.; Wang, N.; Liu, T.; and Gao, X. 2024. Task-Aware Orthogonal Sparse Network for Exploring Shared Knowledge in Continual Learning. In *International Conference on Machine Learning*.

Huang, W.-C.; Chen, C.-F.; and Hsu, H. 2024. OVOR: OnePrompt with Virtual Outlier Regularization for Rehearsal-Free Class-Incremental Learning. In *International Conference on Learning Representations*.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1): 79–87.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S. J.; Hariharan, B.; and Lim, S.-N. 2022. Visual Prompt Tuning. In *Proceedings of the European Conference on Computer Vision*, volume 13693, 709–727.

Jung, D.; Han, D.; Bang, J.; and Song, H. 2023. Generating Instance-Level Prompts for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11813–11823.

Khan, M. G. Z. A.; Naeem, M. F.; Van Gool, L.; Stricker, D.; Tombari, F.; and Afzal, M. Z. 2023. Introducing Language Guidance in Prompt-Based Continual Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11429–11439.

Kong, Y.; Liu, L.; Wang, Z.; and Tao, D. 2022. Balancing Stability and Plasticity Through Advanced Null Space in Continual Learning. In *Proceedings of the European Conference on Computer Vision*, volume 13686, 219–236.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Kurniawan, M. R.; Song, X.; Ma, Z.; He, Y.; Gong, Y.; Qi, Y.; and Wei, X. 2024. Evolving Parameterized Prompt Memory for Continual Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 13301–13309.

Li, D.; Wang, T.; Chen, J.; Ren, Q.; Kawaguchi, K.; and Zeng, Z. 2024a. Towards Continual Learning Desiderata via HSIC-Bottleneck Orthogonalization and Equiangular Embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 13464–13473.

Li, Z.; Zhao, L.; Zhang, Z.; Zhang, H.; Liu, D.; Liu, T.; and Metaxas, D. N. 2024b. Steering Prototypes with Prompt-Tuning for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2523–2533.

Liang, Y.-S.; and Li, W.-J. 2024. InfLoRA: Interference-Free Low-Rank Adaptation for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23638–23647.

Lin, S.; Yang, L.; Fan, D.; and Zhang, J. 2022. TRGP: Trust Region Gradient Projection for Continual Learning. In *International Conference on Learning Representations*.

Lu, Y.; Zhang, S.; Cheng, D.; Xing, Y.; Wang, N.; Wang, P.; and Zhang, Y. 2024. Visual Prompt Tuning in Null Space for Continual Learning. In *Annual Conference on Neural Information Processing Systems*.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, volume 24, 109–165. Elsevier.

Min, T. D.; Mancini, M.; Alahari, K.; Alameda-Pineda, X.; and Ricci, E. 2023. On the Effectiveness of LayerNorm Tuning for Continual Learning in Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 3577–3586.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment Matching for Multi-Source Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1406–1415.

Qiao, J.; Zhang, Z.; Tan, X.; Chen, C.; Qu, Y.; Peng, Y.; and Xie, Y. 2024. Prompt Gradient Projection for Continual Learning. In *International Conference on Learning Representations*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning*, volume 139, 8748–8763.

Ratcliff, R. 1990. Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions. *Psychological review*, 97(2): 285.

Ridnik, T.; Baruch, E. B.; Noy, A.; and Zelnik, L. 2021. ImageNet-21K Pretraining for the Masses. In *NeurIPS Datasets and Benchmarks*.

Roy, A.; Moulick, R.; Verma, V. K.; Ghosh, S.; and Das, A. 2024. Convolutional Prompting Meets Language Models for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23616–23626.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252.

Rypeść, G.; Cygert, S.; Khan, V.; Trzcinski, T.; Zieliński, B. M.; and Twardowski, B. 2024. Divide and Not Forget: Ensemble of Selectively Trained Experts in Continual Learning. In *ICLR*.

Saha, G.; Garg, I.; and Roy, K. 2021. Gradient Projection Memory for Continual Learning. In *International Conference on Learning Representations*.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q. V.; Hinton, G. E.; and Dean, J. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *ICLR*.

Smith, J. S.; Karlinsky, L.; Gutta, V.; Cascante-Bonilla, P.; Kim, D.; Arbelle, A.; Panda, R.; Feris, R.; and Kira, Z. 2023. 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*, 11909–11919.

Wang, L.; Xie, J.; Zhang, X.; Huang, M.; Su, H.; and Zhu, J. 2023a. Hierarchical Decomposition of Prompt-Based Continual Learning: Rethinking Obscured Sub-optimality. In *Advances in Neural Information Processing Systems*.

Wang, L.; Zhang, X.; Su, H.; and Zhu, J. 2024. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8): 5362–5383.

Wang, R.; Duan, X.; Kang, G.; Liu, J.; Lin, S.; Xu, S.; Lv, J.; and Zhang, B. 2023b. AttriCLIP: A Non-Incremental Learner for Incremental Knowledge Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3654–3663.

Wang, S.; Li, X.; Sun, J.; and Xu, Z. 2021. Training Networks in Null Space of Feature Covariance for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 184–193.

Wang, Y.; Huang, Z.; and Hong, X. 2022. S-Prompts Learning with Pre-trained Transformers: An Occam's Razor for Domain Incremental Learning. In *Advances in Neural Information Processing Systems*.

Wang, Y.; Ma, Z.; Huang, Z.; Wang, Y.; Su, Z.; and Hong, X. 2023c. Isolation and Impartial Aggregation: A Paradigm of Incremental Learning without Interference. In *Proceedings of AAAI*, 10209–10217.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J. G.; and Pfister, T. 2022a. DualPrompt: Complementary Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the European Conference on Computer Vision*, volume 13686, 631–648.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. Learning to Prompt for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 139–149.

Xing, Y.; Wu, Q.; Cheng, D.; Zhang, S.; Liang, G.; Wang, P.; and Zhang, Y. 2023. Dual Modality Prompt Tuning for Vision-Language Pre-Trained Model. *IEEE Transactions on Multimedia*.

Yang, Y.; Cui, Z.; Xu, J.; Zhong, C.; Zheng, W.-S.; and Wang, R. 2023. Continual Learning with Bayesian Model Based on a Fixed Pre-Trained Feature Extractor. *Visual Intelligence*, 1(1): 5.

Yu, J.; Zhuge, Y.; Zhang, L.; Hu, P.; Wang, D.; Lu, H.; and He, Y. 2024. Boosting Continual Learning of Vision-Language Models via Mixture-of-Experts Adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23219–23230.

Zeng, G.; Chen, Y.; Cui, B.; and Yu, S. 2019. Continual Learning of Context-Dependent Processing in Neural Networks. *Nature Machine Intelligence*, 1(8): 364–372.

Zhou, D.-W.; Sun, H.-L.; Ye, H.-J.; and Zhan, D.-C. 2024. Expandable Subspace Ensemble for Pre-Trained Model-Based Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23554–23564.