

# Gradient Projection For Continual Parameter-Efficient Tuning

Jingyang Qiao, Zhizhong Zhang, Xin Tan, Yanyun Qu, Wensheng Zhang, Zhi Han,  
Yuan Xie, Member, IEEE

**Abstract**—Parameter-efficient tunings (PETs) have demonstrated impressive performance and promising perspectives in training large models, while they are still confronted with a common problem: the trade-off between learning new content and protecting old knowledge, leading to zero-shot generalization collapse, and cross-modal hallucination. In this paper, we reformulate Adapter, LoRA, Prefix-tuning, and Prompt-tuning from the perspective of gradient projection, and firstly propose a unified framework called **Parameter Efficient Gradient Projection** (PEGP). We introduce orthogonal gradient projection into different PET paradigms and theoretically demonstrate that the orthogonal condition for the gradient can effectively resist forgetting even for large-scale models. It therefore modifies the gradient towards the direction that has less impact on the old feature space, with less extra memory space and training time. We extensively evaluate our method with different backbones, including ViT and CLIP, on diverse datasets, and experiments comprehensively demonstrate its efficiency in reducing forgetting in class, online class, domain, task, and multi-modality continual settings. The project page is available at <https://dmcv-ecnu-pegp.github.io/>.

**Index Terms**—continual learning, parameter-efficient tuning, anti-catastrophic forgetting, orthogonal gradient projection, multi-modality learning.

## 1 INTRODUCTION

LARGE pre-trained models currently are the prevailing research area in artificial intelligence. In this framework, parameter-efficient tunings (PETs) provide a new perspective for downstream adaption and enable the large pre-trained model to be implemented in specialized domains [18], [19], [26], [27]. However, when continually adapting to consecutive downstream tasks, PETs still exhibit catastrophic forgetting, the phenomenon of learning the current knowledge while concurrently eroding the preceding memory [12], [29], [36], [39], *e.g.*, losing the zero-shot generalization ability or producing hallucination.

In seeking to address the forgetting problem, continual learning, or incremental learning [1], [25], [35], [66] is proposed to train a model with continuously expanded datasets by adding novel classes or domains [3], [6], [20], [23], [34], [45], [48], [55], [63]–[65]. However, traditional continual learning models are unsuitable for fine-tuning large models, *i.e.*, we can't replay or expand as data/model is huge. Additionally, due to the limited connection between distinct PETs, there lacks a unified solution for PETs [33], [50], [59], [60].

- J. Qiao, Z. Zhang, X. Tan and Y. Xie are with School of Computer Science and Technology, East China Normal University, Shanghai, 200062, China; E-mail: 52275901010@stu.ecnu.edu.cn, {zzzhang, xtan, yxie}@cs.ecnu.edu.cn
- Y. Qu is with School of Information Science and Technology, Xiamen University, Fujian, 361005, China; E-mail: yyqu@xmu.edu.cn
- W. Zhang is with Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China; E-mail: wensheng.zhang@ia.ac.cn
- Z. Han is with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China; E-mail: hanzhi@sia.ac.cn

Fortunately, gradient projection (GP), regardless of the backbone network, indicates that continual learning would not forget if the gradient is updated in the orthogonal direction to the subspace spanned by the old features [11], [47], [56]. It modifies the gradient into an orthogonal direction by gradient projection matrix, extracting from sampled feature space. It can provide the anti-forgetting mechanism consuming with fewer extra memory space and training time, which can reduce the burden of fine-tuning large models. However, GP is based on convolutional neural networks [16], leading to the present GP theory cannot be directly applied to parameter-efficient tuning methods. Notice that our prior work has demonstrated that GP theory can be combined with Prompt/Prefix-tuning paradigms [42], while it still lacks a unified anti-forgetting framework with the PET-based methods.

Motivated by the preceding analysis, in this paper, we propose a novel **Parameter Efficient Gradient Projection** (PEGP) method. We recall the pipeline of distinct parameter-efficient tuning (*i.e.*, Prompt-tuning, Prefix-tuning, Adapter, and LoRA) and find that they own a union anti-forgetting equation. Moreover, we solve this anti-forgetting equation by conducting Singular Value Decomposition (SVD) on a sampled feature space. That allows us to obtain the gradient projection matrix in an efficient way.

We empirically investigate the proposed strategy via comprehensive experiments and sensitivity analyses, and demonstrate its generalization ability in various backbones, including (i) ViT and (ii) CLIP. We show that our method substantially outperforms prior state-of-the-art techniques across these backbones, which is observed in Figure 1. Our contributions are summarized as follows:

- Parameter efficient gradient projection is the first unified work to study the anti-forgetting mechanism of

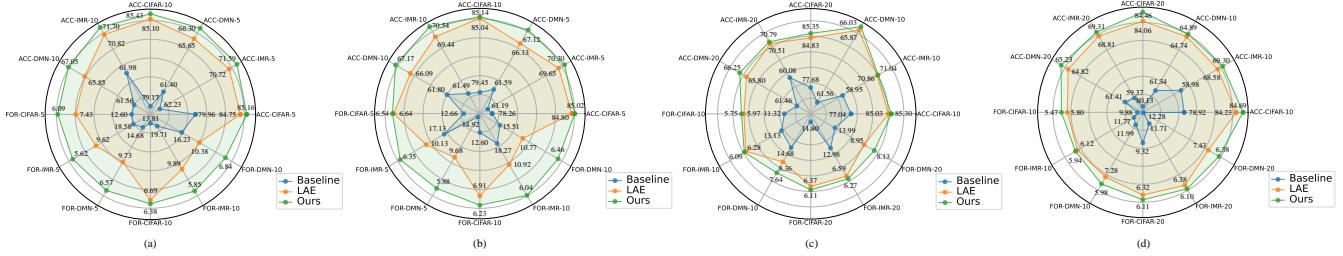


Fig. 1: Radar chart of continual learning results on multiple datasets based on ViT backbone with (a) Adapter (b) LoRA (c) Prefix-Tuning (d) Prompt-Tuning paradigms. ACC refers to the average accuracy metric (higher is better). FOR refers to the forgetting metric (lower is better). An illustrative example of "ACC-CIFAR-10": Average accuracy metric on the 10-Split-CIFAR100 dataset with corresponding tuning parameters of 10 width.

parameter-efficient tuning. Our approach can be applied in various parameter-efficient tuning paradigms, where the forgetting is significantly reduced on most benchmarks.

- Based on the hypothesis that old tasks should have the same results after model updating, our approach obtains the orthogonal condition of gradient and finds the projection matrix towards the direction that has less impact on the old feature space, with less extra memory space and training time.
- Our approach achieves state-of-the-art results in terms of forgetting and average accuracy metrics, under the settings of task, domain, class, online class, and cross-modality incremental learning on diverse datasets.

## 2 RELATED WORK

### 2.1 Continual Learning

Continual learning refers to training deep neural networks (DNNs) on time-variant data [7], [14], [46]. In this context, data is organized into a sequence of tasks denoted as  $\{\mathcal{D} = \mathcal{D}_1, \dots, \mathcal{D}_T\}$ , where the  $t$ -th task  $\mathcal{D}_t = \{(x_i^t, y_i^t)_{i=1}^{n_t}\}$  comprises input sample  $x_i^t \in \mathcal{X}_t$  and its corresponding label  $y_i^t \in \mathcal{Y}_t$ . Upon arrival of a new task  $\mathcal{D}_t$ , a model  $f$  is trained with no access to data from previous tasks.

In detail, it could be divided into *i.e.*, task-, domain-, class- and online-incremental learning, abbreviated as TIL, DIL, CIL, and OIL respectively [15], [52]. Providing a task identifier (abbreviated as task ID) in the inference stage is deemed as task incremental learning, otherwise class incremental learning. Each task maintains the same classes, while the domains of tasks are different from each other, which is recognized as domain incremental learning. Online class incremental learning refers to each training sample only appearing once, in other words, each task is trained with only a single epoch. In this paper, in order to verify the generalization of our method, the above incremental scenarios will be involved sequentially.

### 2.2 PET-Based Continual Learning

PET-based continual learning originated from a simple yet effective Prompt-based CIL model: Learning to Prompt (L2P) [59]. In it, a prompt pool existed, and an instance-wise query mechanism was enabled to pick up suitable prompts. Due to the repeated usage of prompts, newly

learned knowledge would cover old ones, leading to forgetting. To address this problem, the S-Prompts method was proposed, utilizing task-specific prompts that were only trainable in the corresponding task and frozen in other tasks [58]. However, assigning an independent prompt to each task introduces a memory burden as the number of tasks increases, and task identifier inference remains an unsolved challenge. To reduce forgetting while introducing fewer new prompts in each task, based on Prefix-tuning, DualPrompt divided the prompts into two parts: expert prompts (task-specific) and general prompts (task-shared), inserting them into different layers for learning distinct features [60].

One unified parameter-efficient continual method considered four tuning paradigms (Prompt-tuning, Prefix-tuning, Adapter, and LoRA) by a simple and direct Exponential Moving Average (EMA) method to resist forgetting [13]. However, forgetting was not explicitly modeled in this framework, and the mechanism against forgetting has not been revealed yet.

In contrast to the above literature, our work is the first to propose a unified framework to solve the forgetting problem in PET-based continual learning with mathematical demonstration. Its theoretical deduction is in Sec.3 and its effectiveness is verified in Sec.4.

### 2.3 Background of Gradient Projection Method

Gradient limitation, originating from sophisticated mathematical theory, restricts the gradient direction and offers a significant explanation for the stability-plasticity dilemma [4], [32], [68].

Recent research has demonstrated that learning can avoid forgetting if gradients are updated in the direction orthogonal to the subspace spanned by old features [61]. The Gradient Projection Method (GPM), for instance, updates weights in the orthogonal direction to the subspace formed by previously learned inputs, thereby ensuring new learning processes do not disrupt prior tasks [47]. Trust Region Gradient Projection (TRGP) extends this approach by constraining updates within a trust region and employing layer-wise scaling matrices along with orthogonal gradient projection to accommodate new tasks [30]. Simple Linear Connector (Connector) integrates two models using a weighted sum function, allowing one model to update normally while the other updates by using orthogonal gradient projection [31].

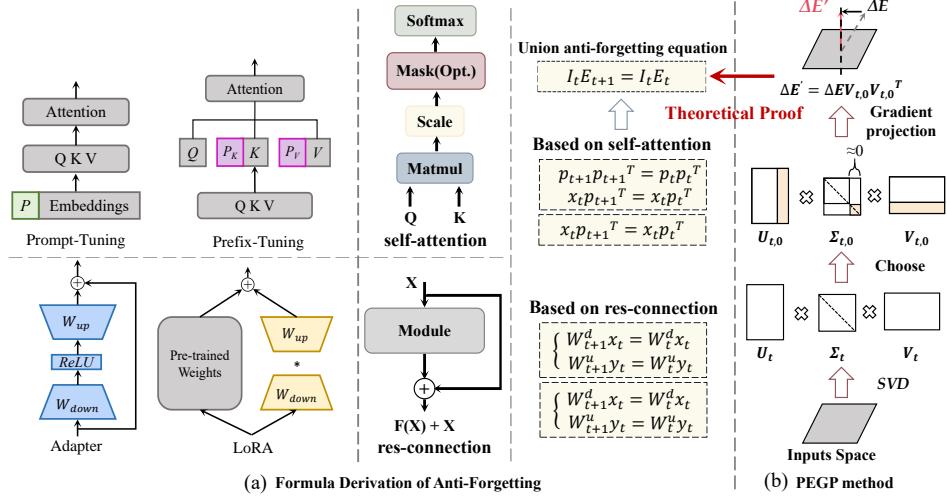


Fig. 2: Illustration of our motivations and methods. (a) Through the investigation of four PETs, we discover a unified anti-forgetting formula from two distinct mechanisms. (b) Implementation of the PEGP process, including feature space sampling, singular value decomposition, gradient projection matrix obtaining, and gradient projection.

However, one limitation of gradient projection methods is only applicable to convolutional neural networks and lack a theoretical foundation in parameter-efficient tuning, which is based on vision transformers. In this paper, we will prove that gradient projection methods can aid in resisting forgetting for parameter-efficient tuning, and the combination of these two methods shows advanced properties in continual learning through experiments.

## 2.4 Differences From the Preliminary Version

This study is a journal extension of the conference paper with the following differences and improvements [42]:

(1) The motivations and formulations are different from our prior work [42]. Our previous work primarily provides specific deductions and solutions to alleviate forgetting in Prompt/Prefix-tuning. In contrast, PEGP proposes a unified framework that can simultaneously solve the problem in total four parameter-efficient tuning paradigms (Adapter/LoRA/Prompt/Prefix-tuning). Additionally, PEGP migrates from single modality model, ViT, to cross-modality model, CLIP.

(2) The prerequisites are looser. In our prior work, we extra introduced key-query mechanism to provide task-identifier, which can be an aid in updating proper prompts with projected gradient. In this work, we discover that the gradient projection method can also work well without task identifier.

(3) We extensively evaluate PEGP across diverse continual learning scenes and provide comprehensive ablations isolating each key component. Notice that, we first construct a cross-modality continual learning benchmark called BITM, which is the task of image-text matching. Experiments across all scenarios verify both the theoretical soundness and practical effectiveness of our method.

## 3 METHOD

### 3.1 A Unified Parameter-Efficient Continual Method

In our gradient projection framework, parameter-efficient tunings could be categorized as two sub-types: 1) Prompt/Prefix-tuning based on **self-attention** as prepended vectors interacting with inputs in self-attention calculation and 2) Adapter/LoRA based on **res-connection**, due to adding bypass in the net module. Although starting from distinct insertion, we surprisingly find that the forward of these modules seems similar and the anti-forgetting character can be described by a union equation. Here, we take class incremental learning as an example, and suppose we have trained on the  $t$ -th task and met the next  $t + 1$ -th task. The old inputs are denoted as  $x_t$  and the new/old trainable parameters as  $E_{t+1}$ ,  $E_t$  respectively. Our motivation and implementation method are shown in Figure 2.

To better preserve old knowledge, we propose that in the ideal state of anti-forgetting, the update of the network would satisfy the following proposition:

**Proposition 1.** Starting from the old inputs from previous tasks  $x_t$  have the same outputs after learning a new task, we have:

$$f_\theta(E_{t+1}, x_t) = f_\theta(E_t, x_t), \quad (1)$$

where  $f_\theta$  refers to the backbone parameters. Unrelated with the specific tuning paradigm, forgetting is avoided if we can have the following equation (Detailed proof is in the following subsections):

$$x_t E_{t+1} = x_t E_t. \quad (2)$$

Expand  $E_{t+1}$  as  $E_{t+1} = E_t + \Delta E$  and remove the identical items in each side of the union equation. Finally, we can obtain:

$$x_t \Delta E = 0. \quad (3)$$

Eq.(3) implies that if the gradient is updated in the orthogonal direction to the subspace spanned by old features, the forgetting would be significantly reduced.

**Discussion:** Compared with previous works, we firstly propose a union anti-forgetting method based on parameter efficient tunings. Please kindly refer to C in the supplementary materials for the detailed algorithm process. Our method is independent of the specific tuning mechanism and owns strong generalization ability without any assumptions.

### 3.2 Self-Attention Based Gradient Projection Method

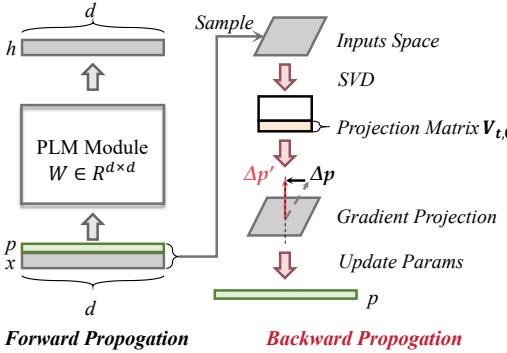


Fig. 3: Flowchart for Prompt-based gradient projection.

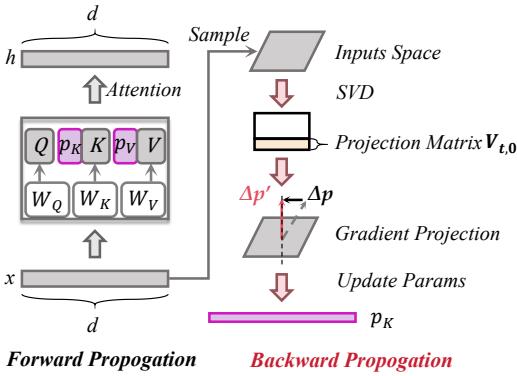


Fig. 4: Flowchart for Prefix-based gradient projection.

Prompt-tuning prepends prompt vector  $P \in R^{l \times d}$  ( $l$  denotes the sequence length and  $d$  denotes the embedding dimension) with the input embeddings  $x_t$  in the first layer [26]. In this framework, after training task  $t + 1$ , we concatenate the prompts  $p_{t+1}$  and the embedding sequences  $x_t$ , i.e., inputs from  $t$ -th task, along the embedding dimension:  $Z_t^{t+1} = \begin{bmatrix} p_{t+1} \\ x_t \end{bmatrix}$ . Thus the self-attention is conducted based on embedding  $Z$ , where query and key can be represented as  $Q_t^{t+1} = W_q Z_t^{t+1}$  and  $K_t^{t+1} = W_k Z_t^{t+1}$  respectively, which is shown in Figure 3.

Similar to Prompt-tuning, Prefix-tuning prepends the prompt vectors  $p \in R^{L_p \times D}$  with key  $K$  and value  $V$  as  $[p^k; K]$  and  $[p^v; V]$  respectively [27]. Assuming that a set of prefixes have been trained at task  $t + 1$ , we input samples from task  $t$  and have  $Q_t^{t+1} = W_q x_t$ ,  $K_t^{t+1} = \begin{bmatrix} p_{t+1}^k \\ W_k x_t \end{bmatrix}$ , which is shown in Figure 4.

In order to realize Proposition 1, we start from the implementation of the self-attention matrix [9], which is calculated as:

$$A_t^{t+1} = \text{softmax}\left(\frac{Q_t^{t+1} K_t^{t+1}{}^T}{\sqrt{\left(\frac{d}{h}\right)}}\right). \quad (4)$$

Due to the normalized denominator, we mainly focus on the numerator part  $Q_t^{t+1} K_t^{t+1}{}^T$ . In the next, we will discuss two cases separately.

**Prompt-tuning:** The numerator in Eq.(4) can be further expanded as  $W_q Z_t^{t+1} Z_t^{t+1}{}^T W_k^T$ . Because we freeze the weights of visual encoder,  $W_q$  and  $W_k$  are unchanged during training. The parameters influenced by training are  $Z_t^{t+1} \cdot Z_t^{t+1}{}^T$ .

To realize Proposition 1, i.e., the condition of anti-forgetting, we need to achieve  $Z_t^{t+1} \cdot Z_t^{t+1}{}^T = Z_t^t \cdot Z_t^t{}^T$  and the requirements are:

$$\begin{cases} p_{t+1} p_{t+1}^T = p_t p_t^T, \\ x_t p_{t+1}^T = x_t p_t^T, \\ p_{t+1} x_t^T = p_t x_t^T. \end{cases} \quad (5)$$

Please kindly refer to A.1 in the supplementary materials for the whole demonstration process of Eq.(5).

**Prefix-tuning:** By further expanding the numerator part  $Q_t^{t+1} K_t^{t+1}{}^T$ , we have the equation that satisfy Proposition 1 as:

$$x_t p_{t+1}^T = x_t p_t^T, \quad (6)$$

Please kindly refer to A.2 in the supplementary materials for the whole demonstration process of Eq.(6).

In the following content, we take the Prompt-tuning as an example. In Eq.(5),  $p_{t+1}$  is divided into  $p_t$  and  $\Delta p$ , where  $\Delta p$  is the gradient of prompts when training task  $t + 1$ . Thus we extend  $p_{t+1} p_{t+1}^T$  as:

$$p_{t+1} p_{t+1}^T = (p_t + \Delta p)(p_t + \Delta p)^T \quad (7)$$

$$= p_t p_t^T + p_t \Delta p^T + \Delta p p_t^T + \Delta p \Delta p^T. \quad (8)$$

Here we ignore the high-order infinitesimal term of  $\Delta p \Delta p^T$ . By observation, if  $p_t \Delta p^T = 0$ , the first term in Eq.(5) i.e.,  $p_{t+1} p_{t+1}^T = p_t p_t^T$  can be realized.

Similarly, the second term in Eq.(5) can be expanded as:

$$x_t p_{t+1}^T = x_t (p_t^T + \Delta p^T) = x_t p_t^T + x_t \Delta p^T = x_t p_t^T. \quad (9)$$

Eliminating  $x_t p_t^T$  on both sides, we have  $x_t \Delta p^T = 0$ . Note that this condition also satisfies the third term in Eq.(5) because  $x_t p_{t+1}^T$  is the transpose of  $p_{t+1} x_t^T$ .

Therefore, we reach the key conclusion: antiforgetting ability can be obtained by restricting the gradient of prompts to satisfy the following equations:

$$\begin{cases} x_t \Delta p^T = 0, \\ p_t \Delta p^T = 0. \end{cases} \quad (10)$$

To solve Eq.(10), we take the first term as an example, we decompose  $x_t$  with SVD:  $x_t = U_t \Sigma_t V_t^T$ . Here,  $U_t$  and  $V_t$

1. Here we omit the factor of learning rating since this simplification wouldn't influence our conclusion.

contain singular vectors corresponding to singular values in  $\Sigma_t$ , and matrix  $\Sigma_t$  can be further divided as:

$$\Sigma_t = \begin{bmatrix} \Sigma_{t,1} & O \\ O & \Sigma_{t,0} \end{bmatrix}, \quad (11)$$

where  $\Sigma_{t,1}$  denotes the non-zero singular values of  $\Sigma_t$  and  $\Sigma_{t,0}$  represents the near-zero singular values of  $\Sigma_t$  [8]. Corresponding to the value of element in  $\Sigma_t$  whether equals to zero,  $V_t$  can be divided into two parts along the column dimension:  $V_t = [V_{t,1}, V_{t,0}]$ . Thus, we have:

$$x_t[V_{t,1}, V_{t,0}] = U_t \begin{bmatrix} \Sigma_{t,1} & O \\ O & \Sigma_{t,0} \end{bmatrix}. \quad (12)$$

Finally, we have the equation:

$$x_t V_{t,0} = U_t \begin{bmatrix} O \\ \Sigma_{t,0} \end{bmatrix} \approx O. \quad (13)$$

Let  $\Delta p = \Delta p V_{t,0} V_{t,0}^T$ , we can obtain:

$$x_t \Delta p^T = x_t (\Delta p V_{t,0} V_{t,0}^T)^T = x_t V_{t,0} V_{t,0}^T \Delta p^T = O. \quad (14)$$

By taking  $V_{t,0}$  as the gradient projection matrix, we successfully realize the first term in Eq.(10).

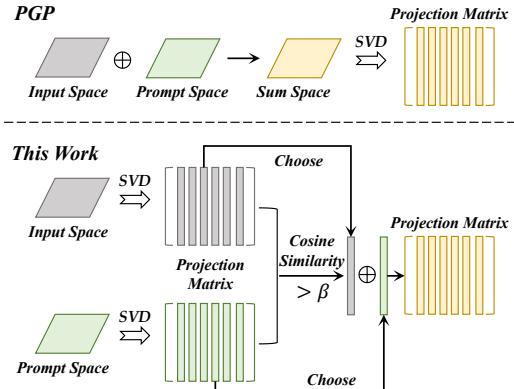


Fig. 5: Comparison of gradient projection matrix obtaining method between PGP and this work.

In our previous work, we sum the prompt space and inputs space directly to create the sum space [42], on which we calculate the gradient projection matrix. However, the summation strategy would fail if the column vectors from each space own totally different directions, influencing the result of gradient projection.

In this paper, we calculate the cosine similarity between the column vectors of  $V_{t,0}^i$  and  $V_{t,0}^p$  in the same location, where  $V_{t,0}^i$  denotes the projection matrix of inputs and  $V_{t,0}^p$  refers to the projection matrix of prompt. Then we compare the similarity score with a hyperparameter threshold  $\beta$ . If the score is larger *i.e.*, the two column vectors have similar directions, we sum the two column vectors and put the result into the final projection matrix  $V_{t,0}$ . Otherwise, we continue to compare the residual column vectors. The whole process is shown in Figure 5.

### 3.3 Res-Connection Based Gradient Projection Method

Adapter includes a down-sampling layer with weights  $W^d \in R^{d \times r}$  to reduce dimension from a higher dimension

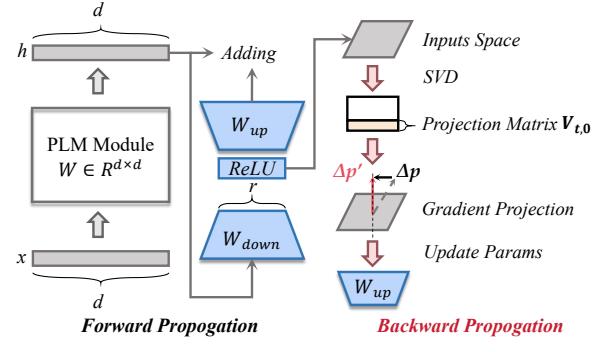


Fig. 6: Flowchart for Adapter-based gradient projection.

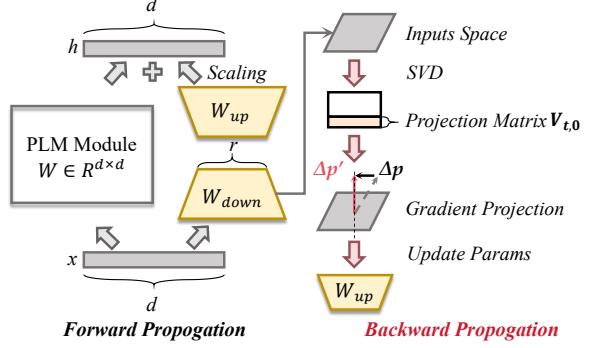


Fig. 7: Flowchart for LoRA-based gradient projection.

$d$  to a lower dimension  $r$ , followed by a nonlinear activation function  $f(\cdot)$ , and a up-sampling layer with weights  $W^u \in R^{r \times d}$  to restore dimension from  $r$  to  $d$  [18], which is shown in Figure 6. In each Adapter module, inputs  $x$  and outputs  $h$  can be described as:

$$h = f_\theta(x) + f_{act}(W^u W^d x). \quad (15)$$

LoRA prepends tiny trainable low-rank matrices  $W^d \in R^{d \times r}$  and  $W^u \in R^{r \times d}$  into backbone parameters as  $W + \Delta W = W + W^u W^d$  [19], which is shown in Figure 7. In LoRA module, inputs  $x$  and outputs  $h$  can be described as:

$$h = f_\theta(x) + s \cdot W^u W^d x. \quad (16)$$

From Proposition 1, if we eliminate identical terms on both sides of these two equations respectively, we have:

$$\begin{cases} f_{act}(W^u W^d x_t) = f_{act}(W_{t+1}^u W_{t+1}^d x_t), \\ s \cdot W_t^u W_t^d x_t = s \cdot W_{t+1}^u W_{t+1}^d x_t. \end{cases} \quad (17)$$

In addition to the scalar  $s$  and activation function  $f_{act}$ , we can find that the mathematical form of Adapter is highly similar to LoRA. Thus, we can both obtain the following equation to resist the forgetting in both paradigms:

$$W_t^u W_t^d x_t = W_{t+1}^u W_{t+1}^d x_t. \quad (18)$$

Define the update of  $W^d$ ,  $W^u$  as  $\Delta W^d$  and  $\Delta W^u$ . We can present the  $W_{t+1}^d$  and  $W_{t+1}^u$  by  $W_t^d$  and  $W_t^u$  as:

$$\begin{cases} W_{t+1}^d = W_t^d + \Delta W^d, \\ W_{t+1}^u = W_t^u + \Delta W^u. \end{cases} \quad (19)$$

By substituting Eq.(19) into Eq.(18), we can obtain:

$$W_t^u W_t^d x_t = (W_t^d + \Delta W^d)(W_t^u + \Delta W^u)x_t. \quad (20)$$

By rearranging the above equations, it can be derived from:

$$\begin{cases} W_{t+1}^d x_t = (W_t^d + \Delta W^d)x_t, \\ W_{t+1}^u y_t = (W_t^u + \Delta W^u)y_t, \end{cases} \quad (21)$$

where  $y_t = W_t^d x_t = W_{t+1}^d x_t$ . Removing the identical items in the both sides of Eq.(21), we can yield:

$$\begin{cases} \Delta W^d x_t = 0, \\ \Delta W^u y_t = 0. \end{cases} \quad (22)$$

Therefore, Proposition 1 is shifted to Eq.(22), where two equations have the same form.

To achieve Eq.(22), taking  $\Delta W^d x_t = 0$  as an example, we perform singular value decomposition and obtain  $SVD(x_t) = U_t \Sigma_t V_t^T$ . Further we have  $U_t^T x_t = \Sigma_t V_t^T$ . By selecting only the eigenvectors in  $U_t$  corresponding to zero or close to zero eigenvalues, we can obtain:

$$U_{t,0}^T x_t = \Sigma_{t,0} V_{t,0}^T = \begin{bmatrix} O \\ \Sigma_{t,0} \end{bmatrix} V_{t,0}^T \approx O. \quad (23)$$

Thus, if we choose  $U_{t,0} U_{t,0}^T$  as the projection matrix, then we have:  $\Delta W^d = \Delta W^d U_{t,0} U_{t,0}^T$ . At this point, it holds that:

$$\Delta W^d x_t = \Delta W^d U_{t,0} U_{t,0}^T x_t = 0. \quad (24)$$

Similarly, we can also achieve  $\Delta W^u y_t = 0$ .

## 4 EXPERIMENTS

In this section, we show the proposed PEGP framework could be flexibly applied to various continual learning tasks: (i) Class/Online class Incremental Learning; (ii) Domain Incremental Learning; (iii) Task Incremental Learning; (iv) Cross-modality Incremental Learning with distinct backbones (i) ViT; (ii) CLIP. The quantitative and qualitative results demonstrate the effectiveness of our approach in various circumstances. All experiments are held on NVIDIA RTX 4090 GPUs.

### 4.1 Evaluation Benchmarks and Protocol

In this section, we introduce the adopted benchmark datasets and corresponding evaluation standards.

**10-Split-CIFAR100** [24] contains 20 major categories and 100 subcategories. For each subcategory, it owns 600 images (500 training images and 100 test images). We construct it by evenly splitting the 100 classes into 10 disjoint tasks, and each task has 10 classes.

**10-Split-ImageNet-R** [17] contains art, cartoons, deviantart, et al. 16 renditions of ImageNet classes. ImageNet-R has 200 ImageNet classes with a total of 30,000 images. We split the total 200 classes into 10 disjoint tasks, and each task has 20 classes.

**10-Split-ImageNet100** [40] selects 100 categories from ImageNet1K. The training set contains 600 annotated images for each class, and the validation set contains 100 annotated images for each class. We split the total 100 classes into 10 disjoint tasks, and each task has 10 classes.

**5-Split-DomainNet** [41] is a dataset for domain adaption and domain incremental learning, which has 345 categories and roughly 600,000 images. Images from DomainNet are split into 5 domains. Here, we deem each domain as a task and each task has 345 classes.

We follow the most popular protocol for evaluation, where average accuracy (Simplified as accuracy or Avg. Acc), forgetting, and new task accuracy (Simplified as New Acc) (please kindly refer to B in the supplementary materials for more details).

## 4.2 Implementation Details

**For ViT Backbone:** Consistent with previous works, we use ViT-B/16 pre-trained on ImageNet-21K as our image encoder [9], which is kept frozen during training. We adopt the LAE model, which is a continual learning method containing Adapter/LoRA/Prompt/Prefix-tuning paradigms, and remove the original online/offline dual model design and EMA model fusion mechanism as our baseline. Furthermore, we add the proposed parameter efficient gradient projection method at the training stage to suppress forgetting.

**For CLIP Backbone:** We utilize the model from our previous work [42], and the backbone is ViT-B-16 pre-trained by OpenAI [43]. On the vision side, we only set a single trainable image Prompt/Linear Adapter shared by all tasks. As for the text side, we set trainable text prompt for each class, which is only trained at the corresponding task according to CoOP [71]. Besides that, we add the gradient projection method at the training stage for efficient parameters and improve our gradient projection method by following [31], which linearly merges two models with/without gradient projection method [38].

## 4.3 Class/Online Class/Task Incremental Learning

### 4.3.1 Experiments on ViT Backbone

**Training Settings:** We train the 10-Split-CIFAR100 for 5 epochs and 10-Split-ImageNet-R for 50 epochs with 24 images (resized as 224\*224\*3) in each batch. Adapter/LoRA width is set at 5/10, and Prefix/Prompt length is set at 10/20. The initial learning rate is 0.0028125 for 10-Split-CIFAR100 and 0.00046875 for 10-Split-ImageNet-R, and the decay rate is 0 with Adam optimizer [2], [5], [22].

**Experimental Results:** We compare the performance of our PEGP method with other SOTA methods in Table 1 under the class incremental setting. We observe that PEGP can greatly improve the average accuracy and reduce forgetting compared with the baseline on the four tuning paradigms, demonstrating its effectiveness and generalization ability. It is also noteworthy that with the aid of gradient projection, PEGP outperforms other SOTA methods on the two datasets by +0.33@Avg. ACC on the 10-Split-CIFAR100 benchmark and +0.88@Avg. ACC on the 10-Split-ImageNet-R benchmark.

The same experimental phenomena can also be explored under online class incremental learning in Table 2. Although the training process only allows one epoch for each task [49], our method still performs stronger anti-forgetting ability compared with baseline and others. Figure 8 and Figure 9

TABLE 1: Class incremental learning (*i.e.*, task identifier is unknown at test phase) results of Avg. ACC and Forgetting on 10-Split-CIFAR100 and 10-Split-ImageNet-R with ViT backbone.

Method	Avenue	Paradigm	10-Split-CIFAR100		10-Split-ImageNet-R	
			Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )	Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )
Baseline	-	Adapter-5	79.96	12.60	62.23	18.58
		Adapter-10	79.17	13.81	61.98	19.71
		LoRA-5	78.26	12.66	61.19	17.13
		LoRA-10	79.45	12.60	61.49	18.27
		Prompt-10	78.92	9.88	58.98	11.77
		Prompt-20	80.13	9.32	59.37	11.71
		Prefix-10	77.04	11.32	58.95	13.13
		Prefix-20	77.68	11.80	60.08	12.98
L2P [59]	CVPR'22	Prompt	83.12	7.66	68.38	6.93
DualPrompt [60]	ECCV'22	Prefix	84.59	5.60	68.57	6.29
LAE [13]	ICCV'23	Adapter-5	84.75	7.43	70.72	9.62
		Adapter-10	85.10	6.69	70.82	9.89
		LoRA-5	84.80	6.64	69.65	10.13
		LoRA-10	85.04	6.91	69.44	10.92
		Prompt-10	84.23	5.80	68.58	6.12
		Prompt-20	84.06	6.32	68.81	6.38
		Prefix-10	85.03	5.97	70.86	6.28
		Prefix-20	84.83	6.37	70.51	6.59
PEGP	Ours	Adapter-5	85.16	6.09	71.59	5.62
		Adapter-10	85.43	6.38	71.70	5.85
		LoRA-5	85.02	6.54	70.30	6.35
		LoRA-10	85.14	6.23	70.54	6.04
		Prompt-10	84.69	5.47	69.30	5.94
		Prompt-20	84.46	6.11	69.31	6.10
		Prefix-10	85.35	5.75	71.04	6.09
		Prefix-20	85.30	6.11	70.79	6.27

TABLE 2: Online class incremental learning results of Avg. ACC and Forgetting on 10-Split-CIFAR100 dataset with ViT backbone.

Method	Avenue	Paradigm	10-Split-CIFAR100	
			Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )
Baseline	-	Adapter-5	75.19	19.61
Baseline	-	LoRA-5	76.58	16.00
LAE [13]	ICCV'23	Adapter-5	81.11	11.44
LAE [13]	ICCV'23	LoRA-5	78.30	12.17
Ours	-	Adapter-5	83.77	6.82
Ours	-	LoRA-5	83.21	7.47

show the curves of accuracy and forgetting with the task number increasing on 10-Split-CIFAR100. We observe that on all tasks, the accuracy of our method is always higher than baseline, and forgetting is always lower than baseline with two different tuning paradigms.

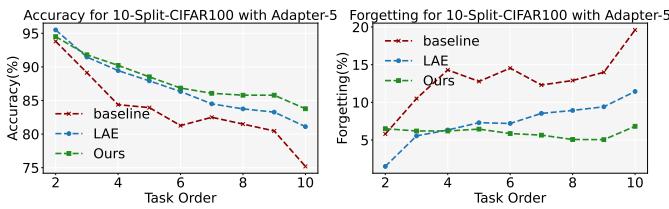


Fig. 8: Task-by-task performance changing curves in terms of accuracy and forgetting under online class incremental learning on 10-Split-CIFAR100 with Adapter-5.

**Ablation Study:** Based on the Figure 10, we can draw similar conclusions in Adapter/LoRA tuning case: With

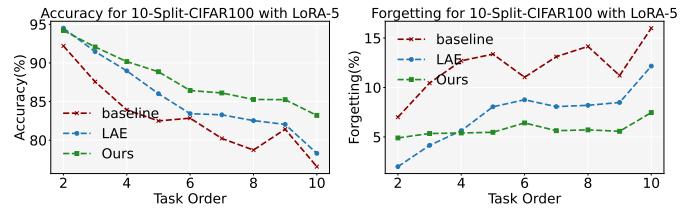


Fig. 9: Task-by-task performance changing curves in terms of accuracy and forgetting under online class incremental learning on 10-Split-CIFAR100 with LoRA-5.

the change of hyper-parameter  $\epsilon$ , leading to the prompt projection matrix  $V_{t,0}$  with distinct numbers of column vectors. If the  $\epsilon$  becomes smaller, more column vectors would be added into the  $V_{t,0}$ , causing lower new task accuracy (worse plasticity) but less forgetting (better stability). On the contrary, bigger  $\epsilon$  refers to the fewer column vectors in the  $V_{t,0}$ , resulting in more forgetting (worse stability) but higher new task accuracy (better plasticity).

Although here we only show the results with Adapter/LoRA tuning paradigms, similar phenomena are also observed in our previous work with Prompt/Prefix tuning paradigms, and detailed mathematical mechanisms behind them are also included [42]. Thus, We can recognize that the essence of the gradient projection method is a kind of trade-off strategy between plasticity and stability. However, different from other dilemmas [37], it has an optimal solution, which is projecting gradient in the direction orthogonal to the subspace spanned by the old inputs, which can not only own the best ability of anti-forgetting but also have minimal damage to plasticity.

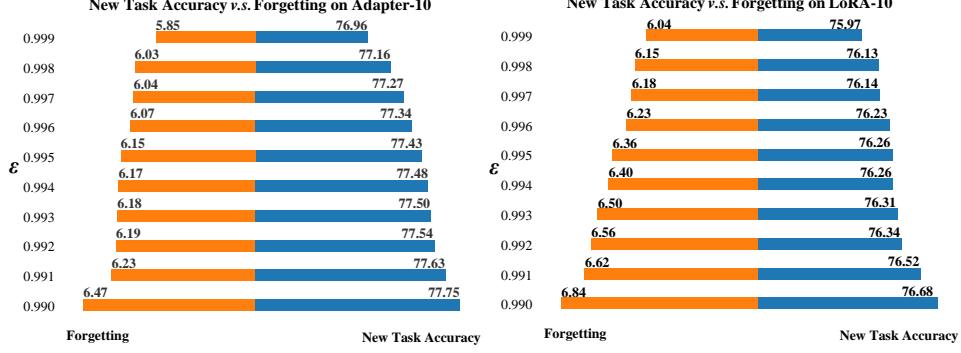


Fig. 10: Ablation study with threshold  $\epsilon$  with Adapter-10/LoRA-10 paradigm on 10-Split-ImageNet-R.

TABLE 3: Class incremental learning results on 10-Split-CIFAR100 and 10-Split-ImageNet100 dataset with CLIP backbone.

Method	Avenue	Paradigm	10-Split-CIFAR100		10-Split-ImageNet100	
			Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )	Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )
Continual-CLIP [51]	ArXiv'22	Prompt	66.70	-	75.40	-
CoOP(Baseline) [71]	IJCV'22	Prompt	73.76	5.60	83.66	2.47
CoOP(Baseline) [71]	IJCV'22	Adapter	63.95	6.28	78.24	5.04
CLIP-EWC [23]	PNAS'17	Prompt	72.29	6.31	81.92	3.58
CLIP-LWF [28]	TPAMI'16	Prompt	75.44	7.42	84.10	2.38
CLIP-PGP [42]	ICLR'24	Prompt	79.47	4.23	84.14	2.11
AttriCLIP [57]	CVPR'23	Prompt	81.40	-	83.30	-
<b>CLIP-PEGP</b>	<b>Ours</b>	<b>Prompt</b>	<b>82.36 (+8.60)</b>	<b>2.82 (-2.78)</b>	<b>85.54 (+1.88)</b>	<b>1.80 (-0.67)</b>
<b>CLIP-PEGP</b>	<b>Ours</b>	<b>Adapter</b>	<b>70.37 (+6.42)</b>	<b>5.72 (-0.56)</b>	<b>79.16 (+0.92)</b>	<b>2.60 (-2.44)</b>

TABLE 4: Task incremental learning results on 10-Split-CIFAR100 and 10-Split-ImageNet100 dataset with CLIP backbone.

Method	Avenue	Paradigm	10-Split-CIFAR100		10-Split-ImageNet100	
			Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )	Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )
CoOP(Baseline) [71]	IJCV'22	Prompt	92.69	2.34	89.08	1.98
CLIP-EWC [23]	PNAS'17	Prompt	<b>94.42</b>	1.22	89.08	1.60
CLIP-LWF [28]	TPAMI'16	Prompt	93.96	1.38	<b>89.36</b>	<b>1.56</b>
CLIP-PGP [42]	ICLR'24	Prompt	93.00	1.58	88.75	1.73
<b>CLIP-PEGP</b>	<b>Ours</b>	<b>Prompt</b>	<b>93.99</b>	<b>0.80</b>	<b>89.08</b>	<b>1.56</b>

#### 4.3.2 Experiments on CLIP Backbone

**Training Settings:** We train the 10-Split-CIFAR100/10-Split-ImageNet100 for 5 epochs with 32 images (resized as 224\*224\*3) in each batch. Linear Adapter width/Prompt length is set at 10. The initial learning rate is 0.001 for the first task and 0.01 for sequential tasks, while the decay rate is 0.1 with the Adam optimizer.

**Experimental Results:** As we compare our method with other SOTA methods based on CLIP backbone, we see that the PEGP can also greatly bring decent improvements both on average accuracy and forgetting, as shown in Table 3. Specifically, on 10-Split-CIFAR100, the method sees an improvement of +0.96 in average accuracy compared with the SOTA method. Similarly, on 10-Split-ImageNet100, the method sees an improvement of surprisingly +1.40 in average accuracy compared with the SOTA method. This validates that the introduction of gradient projection method with parameter-efficient tuning improves model performance without requiring any additional learnable

parameters.

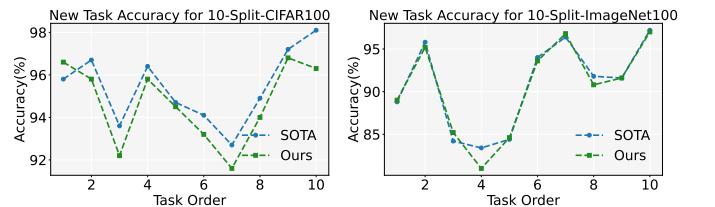


Fig. 11: New task accuracy changing curves with CLIP.

While for task incremental setting, experiments show that our method owns the lowest forgetting but slightly lower accuracy than state-of-the-art on both two datasets in Table 4. Reasons we speculate is that plasticity and stability are trade-off in continual learning. If we focus more on maintaining stability, plasticity is inevitably affected, resulting in a decrease in overall average accuracy. To prove our above hypothesis, we present the new class accuracy

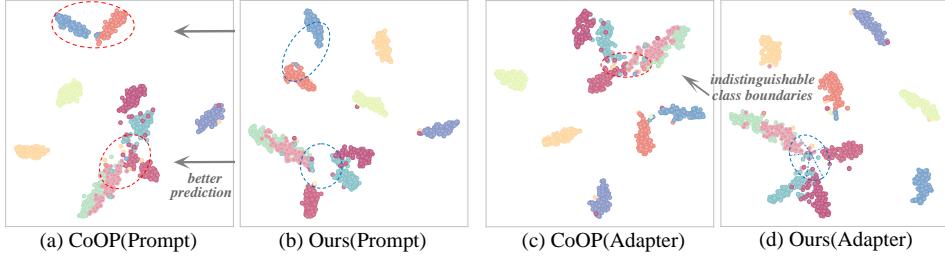


Fig. 12: t-SNE visualization of class incremental learning results on 10-Split-CIFAR100 based on CLIP backbone.

TABLE 5: Domain incremental learning results of Avg. ACC and Forgetting on DomainNet dataset with ViT backbone.

Method	Width	Baseline		LAE [13]		Ours	
		Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )	Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )	Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )
<b>Adapter</b>	5	61.56	14.68	65.65	9.73	<b>66.30</b>	<b>6.57</b>
	10	61.40	16.23	65.85	10.38	<b>67.05</b>	<b>6.84</b>
<b>LoRA</b>	5	61.59	14.92	66.13	9.68	<b>67.12</b>	<b>5.88</b>
	10	61.80	15.51	66.09	10.77	<b>67.17</b>	<b>6.46</b>
<b>Prefix</b>	10	61.56	14.68	65.87	8.36	<b>66.03</b>	<b>7.64</b>
	20	61.46	13.99	65.80	8.95	<b>66.25</b>	<b>8.13</b>
<b>Prompt</b>	10	61.34	11.99	64.74	7.28	<b>64.89</b>	<b>5.98</b>
	20	61.41	12.28	64.82	7.43	<b>65.23</b>	<b>6.38</b>

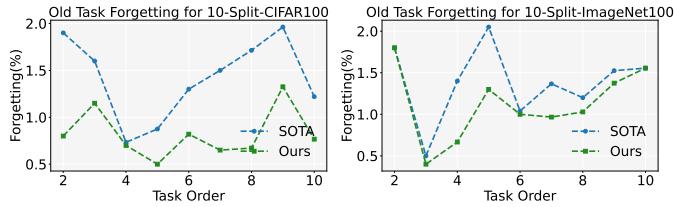


Fig. 13: Old task Forgetting changing curves with CLIP.

changing and old task forgetting curves for each of the 10 tasks, as shown in Figure 11 and Figure 13. It can be seen that the overall new task accuracy curve of our method is always lower than the SOTA method, while old task forgetting is also.

**Visualization:** To better visualize the improvement of our method, we also plot the 2D projection of representations, obtained from the visual output of CLIP. As shown in Figure 12, by using t-SNE, we compare the representations obtained from our approach and baseline. The representations are from task 1 and processed by model trained in task 9. By observing the scatters, we have: (i) the feature space embedded by baseline seems to lack discrimination, where we can observe the overlapping phenomenon between different classes; (ii) the boundaries between some classes of the feature space embedded by baseline looks indistinguishable. However, the above two problems have been well addressed in our method, suggesting better continual learning ability and old knowledge preservation.

**Zero Shot Performance:** Catastrophic forgetting can severely impact the zero-shot generalization ability of large-scale pre-trained models, leading to the zero-shot collapse phenomenon. Based on the proposed PEGP, we design experiments to validate its ability to mitigate zero-shot collapse in pre-trained models. Utilizing CLIP, we first continu-

TABLE 6: Zero-shot performance with CLIP backbone. The column of CLIP refers to the zero-shot performance of original CLIP.

Method	Baseline	PEGP	CLIP
CIFAR100 $\rightarrow$ ImageNet100	70.56	<b>72.84</b>	<b>75.42</b>
ImageNet100 $\rightarrow$ CIFAR100	52.85	<b>59.91</b>	<b>66.72</b>

ally train on the 10-Split-CIFAR100 dataset, followed by testing zero-shot inference ability on the ImageNet100 dataset. Reversely, we continually train on the 10-Split-ImageNet100 dataset and test zero-shot inference ability on the CIFAR100 dataset. The experimental results, as presented in Table 6, demonstrate that PEGP significantly enhances the zero-shot generalization ability of the pre-trained model after continuous fine-tuning in downstream tasks compared to the baseline.

#### 4.4 Domain Incremental Learning

Domain incremental learning aims to verify the continual domain adaption ability of methods. In this setting, images in each task come from distinct domains and the number and kinds of classes in each task should remain the same. This scenario is therefore satisfactory to validate the anti-forgetting effectiveness of our PEGP method with changing domains.

**Training Settings:** We only adopt the ViT backbone in this setting and train the 5-Split-DomainNet for 50 epochs with 24 images (resized as 224\*224\*3) in each batch. Adapter/LoRA width is set at 5/10, and Prefix/Prompt length is set at 10/20. The initial learning rate is 0.00046875, and the decay rate is 0 with the Adam optimizer.

**Experimental Results:** Table 5 shows the comparison results of our PEGP method with baseline and LAE. We

TABLE 7: Cross-modality incremental learning results with CLIP backbone on BITM dataset.

Method	Avenue	Paradigm	10-Split-BITM		5-Split-BITM	
			Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )	Avg. Acc ( $\uparrow$ )	Forgetting ( $\downarrow$ )
CoOP(Baseline) [71]	IJCV'22	Prompt	42.01	33.04	52.73	29.32
CLIP-EWC [23]	PNAS'17	Prompt	50.20	32.62	37.03	25.82
CLIP-LWF [28]	TPAMI'16	Prompt	55.81	23.48	18.31	49.07
CLIP-PGP [42]	ICLR'24	Prompt	54.26	23.01	59.74	24.35
<b>CLIP-PEGP</b>	<b>Ours</b>	<b>Prompt</b>	<b>55.70</b>	<b>22.97</b>	<b>62.66</b>	<b>23.38</b>

observe from Table 5 that PEGP again sets a new state-of-the-art in this setting too. Compared to 5-Split-DomainNet where our method improves SOTA by +1.20@Avg. ACC and -3.54@Forgetting on the Adapter-10 paradigm at the most with the aid of gradient projection, demonstrating its excellent anti-forgetting and generalization ability for the four PET methods.

#### 4.5 Cross-modality Incremental Learning

Cross-modality incremental learning intends to evaluate the effectiveness of the method in continually learning across modalities involving image and text. In this paper, we propose a novel image-text matching dataset called BITM based on the CUB200 dataset [54]. The specific construction method is as follows:

First of all, we select the first 100 classes from the CUB200 dataset. Based on attribute annotations, we extract attributes with certainties of 4 (definitely) and 3 (probably) of each class. These attributes are then sorted based on their frequency of occurrence in images, and the top three attributes with the highest frequency are selected as the text labels. Next, we select images that simultaneously possessed these text labels from the image set. These selected images, along with the text labels, form the text-image matching dataset. We name this new dataset as BITM (Bird Image Text Matching Dataset) and analyze the distribution of text labels and images in the dataset. The visualization of the dataset is shown in the Figure 14. The statistics reveal that BITM contains 100 bird species with a total of 3714 images. The class with the highest number of images contains 59 images, while the class with the lowest number of images contains 13 images. On average, each class contains about 35 images. We divide the dataset into training and testing sets in the ratio of 8:2. Considering that BITM is a continual learning dataset, we further split the dataset into 10-Split-BITM and 5-Split-BITM, where each task contains 10 and 20 classes of images, respectively.

**Benchmarks:** 5/10-Split-BITM are two benchmarks for cross-modality incremental learning. Images from BITM are split into 5/10 disjoint tasks with 20/10 classes in each task.

**Training Settings:** In this setting, we adopt CLIP backbone, incorporating Prompt and Linear Adapter, into the image encoder as baselines. Additionally, in our method, we add PEGP. We train the 5/10-Split-BITM for 30 epochs with 32 images (resized as 224\*224\*3) in each batch. Linear Adapter width/Prompt length is set at 10. The initial learning rate is 0.001 for the first task and 0.01 for sequential tasks, while the decay rate is 0.1 with the Adam optimizer.

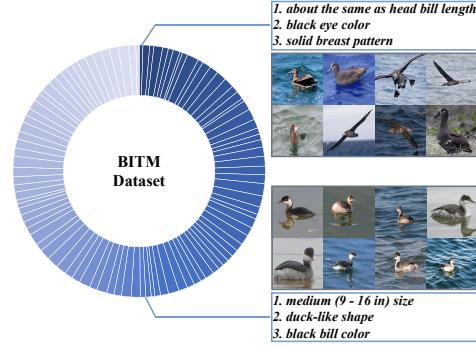


Fig. 14: Visualization of BITM: a novel text-image matching dataset. In the figure, each sector represents a class, and the area of the sector represents the number of images in the corresponding class. The larger the sector, the more images in the corresponding class

Input image	GT	Baseline	Ours
	black eye color all-purpose bill shape yellow underparts color	black eye color yellow underparts color <b>yellow breast color</b>	black eye color all-purpose bill shape <b>yellow underparts color</b>
	the same as head bill length black eye color spatulate bill shape	the same as head bill length plain head pattern <b>hooked seabird bill shape</b>	the same as head bill length black eye color <b>spatulate bill shape</b>
	the same as head bill length black eye color spatulate bill shape	shorter than head bill length black primary color solid back pattern	the same as head bill length black eye color black bill color

Fig. 15: Examples of cross-modality incremental learning results. The red-marked attributes represent the hallucinations in baseline, while the blue-marked attributes represent the predictions corresponding to our method.

**Experimental Results:** We compare the performance of our PEGP method with baseline and other SOTA methods in Table 7. We observe that PEGP can greatly improve the average accuracy and reduce forgetting compared with the baseline on the two benchmarks with distinct tuning paradigms, demonstrating its effectiveness and generalization ability. It is also noteworthy that with the aid of gradient projection, PEGP can not only reduce forgetting but also alleviate the hallucination appearing in the training process. In Figure 15, we can clearly observe that hallucination is present during continually sequential fine-tuning. [67] deem that the hallucination problem in large models is related to the forgetting problem in continual learning, and it can be grouped as external hallucination and internal hallucination. The above view is consistent with our experimental results. However, we are surprised to find that PEGP can

spontaneously suppress the occurrence of hallucinations. We believe that this is inseparable from the anti-forgetting ability of our method. Therefore, we propose that mitigating illusions in large models can be examined through the perspective of reducing forgetting. This discovery may pave the way for addressing the issue of hallucination in large models.

## 5 CONCLUSION

In this work, we theoretically demonstrate that the orthogonal gradient projection can effectively resist forgetting and be applicable to all PET-based continual learning methods. On this basis, we provide a unified framework, Parameter Efficient Gradient Projection, to address the catastrophic forgetting for different tuning paradigms. Additionally, PEGP is proven to be the optimal solution for balancing the trade-off between plasticity and stability in PET-based continual learning methods by choosing the gradient projection matrix. Surprisingly, we also discover that the proposed method aids in overcoming the hallucination problem in the continual learning process. Future works would include learning the large language or multi-modality model to further verify the potential of the gradient projection method. Also, broader learning applications such as image generation or segmentation would be studied.

## APPENDIX

### A. PROOF DETAILS

#### A.1 Proof to Anti-forgetting Equations in Prompt-tuning

In the Prompt-tuning paradigm, to realize Proposition 1, we have the following result from  $Z_t^{t+1} \cdot Z_t^{t+1T}$ :

$$Z_t^{t+1} \cdot Z_t^{t+1T} = \begin{bmatrix} p_{t+1} \\ x_t \end{bmatrix} \begin{bmatrix} p_{t+1}^T & x_t^T \end{bmatrix} = \begin{bmatrix} p_{t+1} p_{t+1}^T & p_{t+1} x_t^T \\ x_t p_{t+1}^T & x_t x_t^T \end{bmatrix}. \quad (25)$$

By contrast, if we replace  $p_{t+1}$  with  $p_t$ , we can obtain the old embedding  $Z_t^t$  through calculation of concatenating prompt trained at task  $t$  and embedding sequences  $x_t$ , and have:

$$Z_t^t \cdot Z_t^{tT} = \begin{bmatrix} p_t \\ x_t \end{bmatrix} \begin{bmatrix} p_t^T & x_t^T \end{bmatrix} = \begin{bmatrix} p_t p_t^T & p_t x_t^T \\ x_t p_t^T & x_t x_t^T \end{bmatrix}. \quad (26)$$

To achieve Proposition 1, *i.e.*, the condition of anti-forgetting, we need to make the equal of Eq.(25) and Eq.(26), and have:

$$\begin{cases} p_{t+1} p_{t+1}^T = p_t p_t^T, \\ x_t p_{t+1}^T = x_t p_t^T, \\ p_{t+1} x_t^T = p_t x_t^T. \end{cases} \quad (27)$$

#### A.2 Proof to Anti-forgetting Equations in Prefix-tuning

With prefixes in key vector and trained at task  $t + 1$ , we input samples from task  $t$  and have:

$$Q_t^{t+1} = W_q x_t, \quad (28)$$

$$K_t^{t+1} = \begin{bmatrix} p_{t+1} \\ W_k x_t \end{bmatrix}, \quad (29)$$

where,  $W_q$  and  $W_k$  are weights of  $i$ -th layer, frozen and unchanged. With Eq.(4), our focus turns to the changing part:

$$\begin{aligned} Q_t^{t+1} K_t^{t+1T} &= W_q x_t [p_{t+1}^T \quad (W_k x_t)^T] \\ &= [W_q x_t p_{t+1}^T \quad W_q x_t x_t^T W_k^T]. \end{aligned}$$

Notice that  $W_q x_t x_t^T W_k^T$  is stable, we only focus on the item  $W_q x_t p_{t+1}^T$ . Changing  $p_{t+1}^T$  with  $p_t^T$ , we can obtain:

$$\begin{aligned} Q_t^t K_t^{tT} &= W_q x_t [p_t^T \quad (W_k x_t)^T] \\ &= [W_q x_t p_t^T \quad W_q x_t x_t^T W_k^T]. \end{aligned}$$

Considering that  $W_q$  is the network parameter, which is frozen, our final goal can be simplified as the following equation to make the equal of  $Q_t^{t+1} K_t^{t+1T}$  and  $Q_t^t K_t^{tT}$ .

$$x_t p_{t+1}^T = x_t p_t^T, \quad (30)$$

## B. EVALUATION METRICS

**Three metrics:** Average Accuracy (Avg. ACC), Forgetting (FOR), and New Task Accuracy (New. ACC) are used to evaluate the performance. We use average accuracy metric, for average test classification accuracy of all tasks. We adopt forgetting metric to indicate the loss of accuracy of past tasks after learning the new task. We employ new task accuracy metric, for average test classification accuracy of new tasks.

$$\text{Average Accuracy} = \frac{1}{T} \sum_{i=1}^T A_{T,i}, \quad (31)$$

$$\text{Forgetting} = \frac{1}{T-1} \sum_{i=1}^{T-1} A_{T,i} - \max(A_{j,i})_{j \in [i, T-1]}, \quad (32)$$

$$\text{New Task Accuracy} = \frac{1}{T} \sum_{i=1}^T A_{i,i}, \quad (33)$$

where  $T$  is the number of tasks,  $A_{T,i}$  is the accuracy of  $i$ -th task samples on the  $T$ -th model,  $A_{j,i}$  is the accuracy of  $i$ -th task samples on the  $j$ -th model, and  $A_{i,i}$  is the accuracy of  $i$ -th task samples on the  $i$ -th model.

## C. ALGORITHM

### REFERENCES

- [1] E. Arani, F. Sarfraz, and B. Zonoz, "Learning fast, learning slow: A general continual learning method based on complementary learning system," *arXiv preprint arXiv:2201.12604*, 2022.
- [2] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing: Algorithms, architectures and applications*. Springer, 1990, pp. 227–236.
- [3] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," *Advances in neural information processing systems*, vol. 33, pp. 15 920–15 930, 2020.
- [4] A. Chaudhry, N. Khan, P. Dokania, and P. Torr, "Continual learning in low-rank orthogonal subspaces," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9900–9911, 2020.
- [5] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, pp. 19–67, 2005.
- [6] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," in *International Conference on Learning Representations*, 2018.

---

**Algorithm 1: Gradient Projection Training Process**


---

**Input:** ViT model  $f_\theta$ , classifier  $f_{c'}$ , number of tasks  $T$ , training set  $\{\{x_i^t, y_i^t\}_{i=1}^{n_t}\}_{t=1}^T$ , sampling set  $\{\{x_{si}^t, y_{si}^t\}_{i=1}^{n_{st}}\}_{t=1}^T$ , efficient parameters  $p$ , number of training epochs  $E$ , projection matrix  $V_{t,0}$ , cross-entropy loss  $\mathcal{L}_{CE}$ , learning rate  $\eta$

**initialize:**  $f_c, p$

**for**  $t = 1, \dots, T$  **do**

- for**  $e_1 = 1, \dots, E$  **do**

  - 1: Draw a mini-batch  $B = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$
  - 2: Obtain batch loss  $\mathcal{L}_B$  by accumulating  $\mathcal{L}_{CE}(y_i^t, f_c(f_\theta(p, x_i^t)))$

- end**
- # Parameter Efficient Gradient Projection
- for**  $t > 1$  **do**

  - 3. Update  $p$  by  $p \leftarrow p - \eta \nabla_p \mathcal{L}_B V_{t,0} V_{t,0}^T$  with Eq.(14) or Eq.(24).

- end**
- # Gradient Projection Matrix Update
- 4. Initialize the sets of sampled features:  $X_t = \{\}$ .
- for**  $(x, y)$  in  $\{(x_{si}^t, y_{si}^t)\}_{i=1}^{n_{st}}$  **do**

  - 5. Sample set of features  $X_t$  from  $f_\theta(p, x)$ .
  - 6. Update  $V_{t,0}$  by  $X_t$  according to [42].

- end**

**end**

---

- [7] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.
- [8] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.
- [9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.
- [10] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "Dytox: Transformers for continual learning with dynamic token expansion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9285–9295.
- [11] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal gradient descent for continual learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 3762–3773.
- [12] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.
- [13] Q. Gao, C. Zhao, Y. Sun, T. Xi, G. Zhang, B. Ghanem, and J. Zhang, "A unified continual learning framework with general parameter-efficient tuning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11483–11493.
- [14] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, "Embracing change: Continual learning in deep neural networks," *Trends in cognitive sciences*, vol. 24, no. 12, pp. 1028–1040, 2020.
- [15] J. He, R. Mao, Z. Shao, and F. Zhu, "Incremental learning in online scenario," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13926–13935.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [17] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. 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*, 2021, pp. 8340–8349.
- [18] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.
- [19] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2021.
- [20] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [21] M. G. Z. A. Khan, M. F. Naeem, L. Van Gool, D. Stricker, F. Tombari, and M. Z. Afzal, "Introducing language guidance in prompt-based continual learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11463–11473.
- [22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [23] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
- [24] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [25] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? complementary learning systems theory updated," *Trends in cognitive sciences*, vol. 20, no. 7, pp. 512–534, 2016.
- [26] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 3045–3059.
- [27] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4582–4597.
- [28] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.
- [29] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong, "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting," in *International conference on machine learning*. PMLR, 2019, pp. 3925–3934.
- [30] S. Lin, L. Yang, D. Fan, and J. Zhang, "Trgp: Trust region gradient projection for continual learning," in *The Tenth International Conference on Learning Representations*, 2022.
- [31] G. Lin, H. Chu, and H. Lai, "Towards better plasticity-stability trade-off in incremental learning: A simple linear connector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 89–98.
- [32] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017.
- [33] Z. Ma, X. Hong, B. Liu, Y. Wang, P. Guo, and H. Li, "Remind of the past: Incremental learning with analogical prompts," *arXiv preprint arXiv:2303.13898*, 2023.
- [34] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.
- [35] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory," *Psychological review*, vol. 102, no. 3, p. 419, 1995.
- [36] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.
- [37] M. Mermilliod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers in psychology*, vol. 4, p. 54654, 2013.
- [38] S. I. Mirzadeh, M. Farajtabar, D. Gorur, R. Pascanu, and H. Ghasemzadeh, "Linear mode connectivity in multitask and continual learning," in *International Conference on Learning Representations*, 2020.

- [39] C. V. Nguyen, A. Achille, M. Lam, T. Hassner, V. Mahadevan, and S. Soatto, "Toward understanding catastrophic forgetting in continual learning," *arXiv preprint arXiv:1908.01091*, 2019.
- [40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [41] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1406–1415.
- [42] J. Qiao, Z. Zhang, X. Tan, C. Chen, Y. Qu, Y. Peng, and Y. Xie, "Prompt gradient projection for continual learning," in *The Twelfth International Conference on Learning Representations*, 2023.
- [43] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [44] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, "Encoder based lifelong learning," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1320–1328.
- [45] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.
- [46] M. B. Ring, "Child: A first step towards continual learning," *Machine Learning*, vol. 28, pp. 77–104, 1997.
- [47] G. Saha, I. Garg, and K. Roy, "Gradient projection memory for continual learning," in *International Conference on Learning Representations*, 2020.
- [48] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International conference on machine learning*. PMLR, 2018, pp. 4548–4557.
- [49] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.
- [50] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, and Z. 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*, 2023, pp. 11909–11919.
- [51] V. Thengane, S. Khan, M. Hayat, and F. Khan, "Clip model is an efficient continual learner," *arXiv preprint arXiv:2210.03114*, 2022.
- [52] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *arXiv preprint arXiv:1904.07734*, 2019.
- [53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [54] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.
- [55] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Foster: Feature boosting and compression for class-incremental learning," in *European conference on computer vision*. Springer, 2022, pp. 398–414.
- [56] S. Wang, X. Li, J. Sun, and Z. Xu, "Training networks in null space of feature covariance for continual learning," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2021, pp. 184–193.
- [57] R. Wang, X. Duan, G. Kang, J. Liu, S. Lin, S. Xu, J. Lü, and B. Zhang, "Attriclip: A non-incremental learner for incremental knowledge learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3654–3663.
- [58] Y. Wang, Z. Huang, and X. Hong, "S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5682–5695, 2022.
- [59] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.
- [60] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy *et al.*, "Dualprompt: Complementary prompting for rehearsal-free continual learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 631–648.
- [61] M. Wortsman, M. C. Horton, C. Guestrin, A. Farhadi, and M. Rastegari, "Learning neural network subspaces," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11217–11227.
- [62] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 374–382.
- [63] S. Yan, J. Xie, and X. He, "Der: Dynamically expandable representation for class incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3014–3023.
- [64] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *International Conference on Learning Representations*, 2018.
- [65] G. Zeng, Y. Chen, B. Cui, and S. Yu, "Continual learning of context-dependent processing in neural networks," *Nature Machine Intelligence*, vol. 1, no. 8, pp. 364–372, 2019.
- [66] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International conference on machine learning*. PMLR, 2017, pp. 3987–3995.
- [67] Y. Zhai, S. Tong, X. Li, M. Cai, Q. Qu, Y. J. Lee, and Y. Ma, "Investigating the catastrophic forgetting in multimodal large language models," in *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.
- [68] Z. Zhao, Z. Zhang, X. Tan, J. Liu, Y. Qu, Y. Xie, and L. Ma, "Rethinking gradient projection continual learning: Stability/plasticity feature space decoupling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3718–3727.
- [69] D.-W. Zhou, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need," *arXiv preprint arXiv:2303.07338*, 2023.
- [70] D.-W. Zhou, Y. Zhang, J. Ning, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Learning without forgetting for vision-language models," *arXiv preprint arXiv:2305.19270*, 2023.
- [71] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.