

# Dynamic Multi-Layer Null Space Projection for Vision-Language Continual Learning

Borui Kang<sup>1</sup>, Lei Wang<sup>2</sup>, Zhiping Wu<sup>1</sup>, Tao Feng<sup>3</sup>, Yawen Li<sup>4</sup>, Yang Gao<sup>1,5</sup>, Wenbin Li<sup>1\*</sup>

<sup>1</sup>State Key Laboratory for Novel Software Technology, Nanjing University, China

<sup>2</sup>University of Wollongong, Australia <sup>3</sup>Tsinghua University, China

<sup>4</sup>Beijing University of Posts and Telecommunications, China

<sup>5</sup>Yili Normal University, Xinjiang, China

## Abstract

Vision-Language Models (VLM) have emerged as a highly promising approach for Continual Learning (CL) due to their powerful generalizable features. While adapter-based VLM can exploit both task-specific and task-agnostic features, current CL methods have largely overlooked the distinct and evolving parameter distributions in visual and language modalities, which are found crucial for effectively mitigating catastrophic forgetting. In this study, we find that the visual modality experiences a broader parameter distribution and greater variance during class increments than the textual modality, leading to higher vulnerability to forgetting. Consequently, we handle the branches of the two modalities asymmetrically. Specifically, we propose a Dynamic Multi-layer Null Space Projection (DMNSP) strategy and apply it only to the visual modality branch, while optimizing the language branch according to the original optimizer. DMNSP can restrict the update of visual parameters within the common subspace of multiple null spaces, further limiting the impact of non-zero residual terms. Simultaneously, combined with a dynamic projection coefficient, we can precisely control the magnitude of gradient projection to the null space, endowing the model with a good balance of stability and plasticity. Extensive experiments on TinyImageNet, CIFAR100 and ImageNet-R demonstrate that our method outperforms current approaches in accuracy and knowledge retention, setting a new standard for state-of-the-art performance in class incremental learning. Our code is available at <https://github.com/RL-VIG/DMNSP>.

## 1. Introduction

The primary objective of Continual Learning (CL) is to enable models to learn from a sequence of data or tasks

\*Corresponding Author

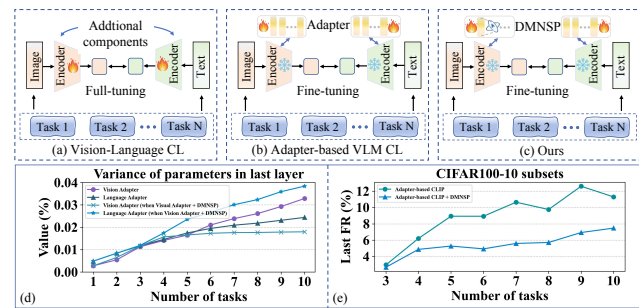


Figure 1. (a) VLM-based CL methods that require full fine-tuning. (b) Adapter-based VLM CL methods requiring fine-tuning of adapter layers. (c) Adapter-based VLM CL methods with DMNSP in visual branch. (d) Variances of visual and language parameters in adapter-based CLIP within CL. (e) Forgetting rate of adapter-based CLIP before and after the integration of DMNSP.

over time while effectively preventing catastrophic forgetting [8, 9, 32]. However, traditional training-from-scratch models [18, 20] can only obtain the feature representations specific to the current task, and they do not demonstrate good generalization ability when used for continual learning. Hence, in recent years, CL has gradually tended to adopt methods based on Vision-Language Models (VLM), such as the Contrastive Language-Image Pre-training (CLIP) model [27]. The studies in [5, 33] show that CLIP, trained with a large number of image-text pairs through contrastive learning, inherently has a strong continual learning ability, which significantly contributes to the development of CL. Subsequently, several improved methods have been developed. As depicted in Figure 1(a), ZSCL [42] and Mod-X [25] utilize additional components, such as specific datasets and specially-constructed loss functions, to conduct full-tuning of CLIP for CL. To some extent, this affects the expression of the generalizable features of CLIP itself. In contrast, Parameter-Efficient Fine-Tuning (PEFT) techniques [1, 13, 14], such as the adapter-based method, effectively integrate task-specific features obtained from fine-tuning and the task-agnostic generalizable features of VLM.

In addition, as shown in Figure 1(b), MoE4Adapters [40] utilizes an adapter-based VLM architecture, incorporating various adapter designs to alleviate the training burden, thereby achieving strong performance in CL.

However, within the context of VLM under the CL framework, the relationship between these two modalities (*i.e.*, visual and text modalities) remains insufficiently explored. To address this, we investigate the parameter variation of the visual and language adapters in the adapter-based CLIP. As shown in Figure 1(d), during a CL process involving 10 tasks, we observe that the parameter variance of the last layer’s visual adapter undergoes more significant changes than that of the language adapter. This implies that during the CL process, the degree of change in visual adapter parameter distributions is greater than that in the language counterpart, and the corresponding forgetting rate is relatively high (as shown by the green line “Adapter-based CLIP” in Figure 1(e)). When the visual adapter is combined with our strategy to be proposed below, the variance of visual adapter parameters is markedly reduced, and simultaneously the forgetting rate is also substantially decreased (as shown by the blue line “Adapter-based CLIP + DMNSP” in Figure 1(e)). Motivated by this observation, we propose that controlling the variation of visual modality parameters could be a promising approach to mitigating catastrophic forgetting in VLM.

To this end, our work takes the approach of null space projection due to its promising performance in CL, as evidenced by the recent studies [19, 21, 34, 41] revealing that updating gradients within the null space of feature representations to constrain residual terms to zero can effectively mitigate catastrophic forgetting. However, these methods face two fundamental limitations. First, they perform SVD of the feature representations in each layer to estimate the null space. It is approximated by the space spanned by the vectors corresponding to smaller (*i.e.*, not strictly zero) singular values. This inherently retains a small amount of variances of the feature representations, leading to non-zero residual terms (as shown in Figure 4). Second, when applied to VLM, existing approaches [26] equally handle visual-language branches despite their distinct modality characteristics. These observations motivate us to design the *Dynamic Multi-layer Null Space Projection (DMNSP)* method, which leverages comprehensive multi-layer null space information to constrain the projection process. As shown in Figure 1(c), we exclusively apply *DMNSP* to the visual branch only to further constrain its distribution and thus limit non-zero residual terms. Moreover, considering that the magnitude of null space projection directly affects the plasticity-stability trade-off in CL [41], we dynamically adjust the degree of projection by measuring the dynamic similarity between a feature space and its null space. Our contributions can be summarized as follows:

- We propose an asymmetric adapter training approach to mitigate the catastrophic forgetting caused by the different behaviors of two modalities in continual learning faced by adapter-based VLM.
- We propose a multi-layer null space gradient projection strategy and enrich it with a dynamic projection coefficient, thereby effectively reducing the negative impact of non-zero residual terms in the process of CL.
- We conduct eight different experimental settings of Class Incremental Learning (CIL) on the CIFAR100, TinyImageNet and ImageNet-R datasets and achieved state-of-the-art results in terms of accuracy and forgetting rates.

## 2. Related Works

**Traditional Continual Learning** mainly consists of three principal categories of methods. (a) Regularization-based methods [8, 9, 18, 38] alleviate forgetting by imposing penalties on the changes of parameters significant to previous tasks. One drawback of these methods is that the model capacity is fixed, and the penalty loss will cause the model’s plasticity to decrease as old tasks increase. (b) Architecture-based methods involve learning parameters that are tailored for individual tasks. This can be achieved through network expansion [7, 15]. Nevertheless, a drawback of these approaches is that they frequently necessitate a task identity during the inference stage, which may not always be available or practical in real-world applications. (c) Rehearsal-based methods, as exemplified by [12, 28], rely on storing a portion of the past task experiences in a memory and using it for training in conjunction with the current task. However, they are sensitive to the size of the memory.

**VLM-based Continual Learning.** VLM with generalized feature representations are introduced into continual learning to alleviate the deficiencies of traditional methods. The work in [5, 33] shows that the CLIP exhibits astonishing continual learning performance without any fine-tuning. Consequently, CLIP is widely employed as a VLM in continual learning. CLAP4CLIP [17] presents a method that employs variational inference to adapt visual language models to new tasks while preventing forgetting. PROOF [44] utilizes the fusion of multimodal information such as visual language, prototypes, and prompts, combined with task-specific mapping to avoid impacts on old tasks.

However, the aforementioned methods, with an excessive focus on learning task-specific features of new tasks, undermine the expression of old tasks and consequently result in the forgetting phenomenon. Therefore, the adapter-based approach [13] is introduced into the CL of VLM. PEGP [26] explores numerous fine-tuning methods based on VLM, including adapters, to accomplish CL but does not innovate the anti-forgetting method. MoE4Adapters [40] proposes to complete the learning of task-specific features through the mixture of experts [30] and the routing selec-

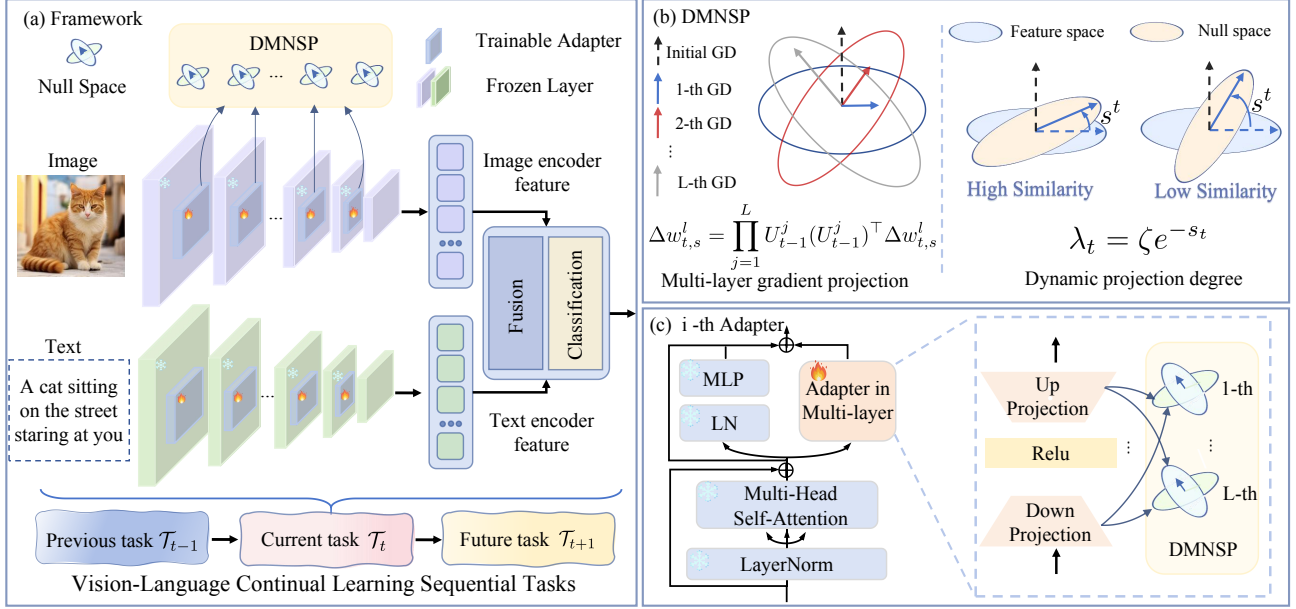


Figure 2. (a) Represents the overall framework of our method. The CLIP backbone network is frozen, and a trainable adapter layer is added to each layer of the vision-language branch. The DMNSP strategy is integrated into the optimization of the visual adapter. (b) Represents the DMNSP strategy, which mainly consists of multi-layer gradient projection and dynamic projection coefficients. The initial gradient is gradually projected onto the null spaces of layers from 1 to L to obtain the common null subspace of all null spaces. Meanwhile, combined with the dynamically changing coefficients, the projection can be better accomplished. (c) Represents the details of the adapter. A combination of two-layer linear layers and a one-layer ReLU function is employed. Only the visual adapter is optimized in conjunction with DMNSP, and the language adapter is optimized using the original optimizer.

tion mechanism, thereby avoiding impacts on old tasks and achieving the current SOTA effect. Nevertheless, due to the lack of a flexible anti-forgetting strategy and the dependence on a fixed adapter-router pair, MoE4Adapters cannot dynamically learn more tasks in class incremental learning. Moreover, existing adapter-based VLM methods have neglected the differential changes in the parameter distribution of visual-language modalities within continual learning tasks. Hence, targeting the dual-branch structure of VLM, we propose an asymmetric optimization method to tackle the problem of catastrophic forgetting.

**Projection-based Continual Learning** constrains the gradient within specific directions for update, thereby avoiding the interference of the new task learning process on old tasks. GPM [29] first projects new gradients onto the subspace important for old tasks and then subtracts the projected components to update parameters. TRGP [22] proposes a “trust region” to select related old tasks via gradient projection for a new task. VPT-NSP<sup>2</sup>[24] uses prompt gradient orthogonal projection to prevent interference with previous tasks and achieve superior anti-forgetting performance. However, when using Singular Value Decomposition (SVD), the above methods do not impose constraints on the residuals generated by the approximation of the null space, and this may compromise the effectiveness of anti-forgetting. Additionally, the gradient projection process

in these methods relies solely on static hyperparameters to control the degree of projection onto the null space [41], which may disrupt the balance between plasticity and stability. These limitations will be addressed by our method proposed in this paper.

### 3. Methodology

In this section, we propose the adapter-based VLM continual learning framework as shown in Figure 2 and elaborate on the details of each part of the method.

#### 3.1. Preliminaries

In continual learning, a network  $f$  with parameters  $\mathbf{w}$  is trained sequentially on a series of tasks  $\mathcal{T}_1, \dots, \mathcal{T}_t$ , where task  $\mathcal{T}_t$  is associated with a paired dataset  $\{X_t, Y_t\}$ . Here,  $X_t$  represents the raw data, and  $Y_t$  denotes the ground truth corresponding to  $X_t$ . When training  $f$  on task  $\mathcal{T}_t$  at the  $s$ -th training iteration, the network parameters are denoted as  $\mathbf{w}_{t,s} = [\mathbf{w}_{t,s}^1, \dots, \mathbf{w}_{t,s}^L]$ , where  $L$  represents the total number of adapter layers. Correspondingly, the parameter update at the  $s$ -th training iteration for task  $\mathcal{T}_t$  is denoted as  $\Delta \mathbf{w}_{t,s} = [\Delta \mathbf{w}_{t,s}^1, \dots, \Delta \mathbf{w}_{t,s}^L]$ , and the gradient is given by  $\mathbf{g}_{t,s} = [\mathbf{g}_{t,s}^1, \dots, \mathbf{g}_{t,s}^L]$ .

### 3.2. Asymmetric adapters

We follow the literature [4] to insert the adapter to each transformer block in CLIP. As shown in Figure 2(c), adapter is a bottleneck structure consisting of a down-projection layer  $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d \times r}$ , a non-linear activation ReLU and an up-projection layer  $\mathbf{W}_{\text{up}} \in \mathbb{R}^{r \times d}$  where  $r \ll d$ . It mainly changes the residual connection in the transformer blocks by adding non-linear transformations to the identical input. For the  $i$ -th block, we denote its output after the multi-head self-attention as  $\hat{\mathbf{x}}_i \in \mathbb{R}^{n \times d}$ , where  $n$  represents the number of patches. We insert adapters to the MLP structure and get the output as:

$$\hat{\mathbf{x}}_{i+1} = \text{MLP}(\text{LN}(\hat{\mathbf{x}}_i)) + \text{ReLU}(\hat{\mathbf{x}}_i \cdot \mathbf{W}_{\text{down}}) \cdot \mathbf{W}_{\text{up}}, \quad (1)$$

where  $\text{MLP}(\text{LN}(\hat{\mathbf{x}}_i))$  is the original output of the transformer block, and LN denotes layer normalization.

We denote the adapters inserted into the visual branch of CLIP as  $A_v$  and the adapters in the language branch as  $A_t$ . Through experiments, we find that during the process of the adapter-based VLM learning on new tasks, the parameter distribution of  $A_v$  is broader than that of  $A_t$ , with more variance changes and a higher forgetting rate. Motivated by this observation, we propose an asymmetric approach to mitigate the significant changes in visual parameters. Specifically, we project the parameters of  $A_v$  onto the multi-layer null space for optimization while optimizing  $A_t$  in the original manner, thereby helping the model to achieve a better balance of stability and plasticity.

### 3.3. Vanilla null space gradient projection

**Vanilla null space.** After the network  $f_t$  completes learning the new task  $\mathcal{T}_t$ , its parameters are updated to  $\mathbf{W}_t$ . When data  $\mathbf{x}_{t-1}$  from the old task  $\mathcal{T}_{t-1}$  is input into the current network, the following condition holds:

$$\mathbf{W}_t \cdot \mathbf{x}_{t-1} = \mathbf{W}_{t-1} \cdot \mathbf{x}_{t-1} + \Delta \mathbf{W}_t \cdot \mathbf{x}_{t-1}. \quad (2)$$

$\mathbf{W}_{t-1} \cdot \mathbf{x}_{t-1}$  represents the feature representation of the old data in previous network  $f_{t-1}$ , while  $\Delta \mathbf{W}_t \cdot \mathbf{x}_{t-1}$  (also known as the residual term) denotes the feature representation of the old data under the new parameter update. When  $\Delta \mathbf{W}_t \cdot \mathbf{x}_{t-1}$  approaches zero, it implies that the parameter changes induced by learning the new task do not impact the data from old tasks [29]. According to the definition of the null space, all the basis vectors  $\mathbf{v}$  of the null space of  $\mathbf{x}_{t-1}$  satisfy  $\mathbf{x}_{t-1} \cdot \mathbf{v} = 0$ . Therefore, we only need to project  $\Delta \mathbf{W}_t$  onto the null space of  $\mathbf{x}_{t-1}$  to avoid the catastrophic forgetting in CL. Specifically, we utilize Singular Value Decomposition (SVD) to extract key information from the uncentered covariance of input features [34] and approximate the null space of this feature by the subspace associated with the singular values of lower magnitudes, denoted as  $\mathbf{U}_{t-1}$ .

In the  $l$ -th adapter, during the current iteration  $s$  of updating the weight matrix  $\Delta \mathbf{w}_{t,s}^l$  for task  $t$ , the null space  $\mathbf{U}_{t-1}^l$  corresponding to this layer is incorporated to adjust the original gradient  $\mathbf{g}_{t,s}^l$  in specific directions, thereby preventing interference with the previous tasks:

$$\Delta \mathbf{w}_{t,s}^l = \mathbf{U}_{t-1}^l (\mathbf{U}_{t-1}^l)^\top \mathbf{g}_{t,s}^l. \quad (3)$$

### 3.4. Multi-layer null space gradient projection

**Multi-layer null space.** We argue that when non-zero residuals exist within local adapters, they will propagate through layers, thereby compromising the accuracy of gradient projections in the null spaces obtained at the subsequent layers. To preemptively mitigate the adverse impact of the residuals propagating from shallow to deeper layers, we tighten the projection constraints and leverage all the null spaces obtained at each layer of the network to collectively constrain the gradient projection process. To achieve this, we propose a novel multi-layer null-space projection method which could effectively suppress the magnitude of the residual terms, driving them toward zero. For more explanations, please refer to Section A of the Supplementary Material. As shown in Figure 2(b), we sequentially project gradient updates into each layer’s null space, ultimately converging on the common null subspace for the final update. Our approach can be formally expressed as:

$$\Delta \mathbf{w}_{t,s}^l = \prod_{j=1}^L \mathbf{U}_{t-1}^j (\mathbf{U}_{t-1}^j)^\top \mathbf{g}_{t,s}^l. \quad (4)$$

**Dynamic projection coefficient.** With the introduction of multi-layer null spaces, it is necessary to consider the relationship between the gradient space of the current task and the null spaces of each layer during the projection process [41]. Recall that we conduct SVD on the uncentered covariance features of the input features. Subsequently, we select the eigenvectors corresponding to the top  $p\%$  of the larger singular values to construct the main feature space  $\bar{\mathbf{U}}_t$  for the current task. Let  $\bar{\mathbf{U}}_t = \{\mathbf{u}^1, \dots, \mathbf{u}^i\}$  and recall that  $\mathbf{U}_{t-1} = \{\mathbf{v}^1, \dots, \mathbf{v}^j\}$  denote the null space obtained for the last task, where  $i = 1, 2, \dots, m$  and  $j = 1, 2, \dots, n$  (with  $m$  and  $n$  being the numbers of basis vectors in  $\bar{\mathbf{U}}_t$  and  $\mathbf{U}_{t-1}$  respectively). Empirically, the cosine similarity between the basis vectors of the main feature space and those of the null space  $\mathbf{U}_{t-1}$  is utilized to characterize the similarity between the two spaces:

$$\gamma_{ij} = \frac{\mathbf{u}^i \cdot \mathbf{v}^j}{\|\mathbf{u}^i\|_2 \cdot \|\mathbf{v}^j\|_2}. \quad (5)$$

We form a set  $C_t = \{\gamma_{ij}\}$  using all the  $\gamma_{ij}$  values. To prevent the accumulation of small similarity values that would lead to an insignificant overall similarity difference, we select the significant values from the entire set of similarity

values to characterize the overall similarity. Consequently, we take the average of the top  $q\%$  largest similarity values in  $C_t$  as the similarity between the current main feature space and the null space, denoted as  $s_t$ .

According to the literature [29], the gradient update direction points to the feature space spanned by the input data. Therefore, we can use  $s_t$  to assess the magnitude of gradient projection onto the null space. As shown in Figure 2(b), when  $s_t$  is large, it indicates that the current optimization direction has less impact on the knowledge of old tasks, so the projection onto the null space can be appropriately weakened to maintain the plasticity of the model. When  $s_t$  is small, it indicates that the current optimization direction has a greater impact on the knowledge of old tasks, so the projection onto the null space can be appropriately strengthened to maintain the plasticity of the model. Therefore, we employ a negative exponential function with a positive range to model this process, obtaining the dynamic projection coefficient:

$$\lambda_t = \zeta e^{-s_t}. \quad (6)$$

Since the gradient undergoes successive multiplicative operations, a numerical overflow problem may occur. Therefore, after the projection between the gradient and each layer’s null space is completed, the numerical range needs to be appropriately adjusted. Specifically, we make this adjustment with a fixed scalar  $\zeta$ . Ultimately, the gradient projection process of the adapter at the  $l$ -th layer is as follows:

$$\Delta \mathbf{w}_{t,s}^l = \prod_{j=1}^L \lambda_t^j \mathbf{U}_{t-1}^j (\mathbf{U}_{t-1}^j)^\top \mathbf{g}_{t,s}^l. \quad (7)$$

**Null space updating.** During the null space update process, the work in the literature [21, 29] first eliminates the basis vectors in the null space of the new task that are similar to those in  $\mathbf{U}_{t-1}$ , then expands the remaining basis vectors to obtain the final null space  $\mathbf{U}_t$ , thereby reducing storage space consumption. However, when considering a multi-layer projection strategy, the basis vectors of each null space must be utilized during the projection process to ensure the accuracy of the final null subspace. Therefore, we employ a simple expansion method to combine the null spaces of each task, aiming to obtain the most comprehensive representation of the null space across all tasks. The process is as follows:  $\mathbf{U}_t^j = [\mathbf{U}_1^j, \mathbf{U}_2^j, \dots, \mathbf{U}_{t-1}^j]$ .

### 3.5. Overall optimization procedure

We present the overall parameter update process of visual adapters in Algorithm 1. When learning the first task, gradient projection is not required (as indicated in line 6). Otherwise, our DMNSP strategy is employed. Specifically, at the  $s$ -th training iteration, the gradient  $\mathbf{g}_{t,s} = [\mathbf{g}_{t,s}^1, \dots, \mathbf{g}_{t,s}^L]$  is projected into the common null subspace across all layers (as mentioned in line 8). The obtained parameter update

---

#### Algorithm 1 DMNSP for continual learning

---

**Inputs:** Datasets  $\{X_t, Y_t\}$  for task  $\mathcal{T}_t \in \{\mathcal{T}_1, \mathcal{T}_2, \dots\}$ ; network  $f(\cdot, \mathbf{w})$  with  $L$  adapter layers; learning rate  $\alpha$ .

**Initialization:** Initialize the parameters of  $L$  visual adapters  $\mathbf{w}$  randomly.

```

1: for task  $\mathcal{T}_t \in \{\mathcal{T}_1, \mathcal{T}_2, \dots\}$  do
2:   while not converged do
3:     Sample a batch  $\{\mathbf{x}, \mathbf{y}\}$  from  $\{X_t, Y_t\}$ .
4:     Compute  $f(\mathbf{x}, \mathbf{w}_{t,s})$ , then get candidate parameter
       update  $\mathbf{g}_{t,s} = [\mathbf{g}_{t,s}^1, \dots, \mathbf{g}_{t,s}^L]$ .
5:     if  $t = 1$  then
6:        $\Delta \mathbf{w}_{t,s}^l = \mathbf{g}_{t,s}^l$ 
7:     else
8:        $\Delta \mathbf{w}_{t,s}^l = \prod_{j=1}^L \lambda_t^j \mathbf{U}_{t-1}^j (\mathbf{U}_{t-1}^j)^\top \mathbf{g}_{t,s}^l$ 
9:     end if
10:     $\mathbf{w}_{t,s+1}^l = \mathbf{w}_{t,s}^l - \text{AdamW}(\alpha \Delta \mathbf{w}_{t,s}^l)$ 
11:     $s = s + 1$ 
12:  end while
13:  if  $t > 1$  then
14:    Get  $\mathbf{U}_t^l = [\mathbf{U}_1^l, \mathbf{U}_2^l, \dots, \mathbf{U}_{t-1}^l]$ 
15:  end if
16: end for

```

---

$\Delta \mathbf{w}_{t,s} = [\Delta \mathbf{w}_{t,s}^1, \dots, \Delta \mathbf{w}_{t,s}^L]$  is input into the AdamW optimizer, thus completing the parameter optimization. After learning task  $\mathcal{T}_t$ , the null space is updated as indicated in line 13-15. Compared with visual adapters, text adapters update parameters as usual and follow the rules of the AdamW optimizer.

## 4. Experiments

### 4.1. Experimental setting

**Datasets.** We evaluate our method on the CIL (Class Incremental Learning) setting. Following [24, 42], experiments are conducted on TinyImageNet, CIFAR100, and ImageNet-R. Specifically, the 100 classes of TinyImageNet are partitioned into  $\{5, 10, 20\}$  subsets with 100 base classes, the 100 classes of CIFAR100 into  $\{10, 20, 50\}$  subsets, and the 200 classes of ImageNet-R into  $\{10, 20\}$  subsets, respectively.

**Implementation details.** We employ the CLIP model with ViT-B/16 [27] as our backbone. Trainable adapter layers are added to each layer of the two branches in CLIP. We utilize the AdamW optimizer. Each task is trained for 4 epochs. For all datasets, the batch size is set to 128, and the learning rate is set to 0.001. Empirically, the eigenvectors corresponding to the top 10% leading eigenvalues after SVD are selected for constructing the main feature representation, and the remaining eigenvectors form the null space representation. All experimental results are the averages obtained by using five different rounds of random

Method	Venue	5 subset		10 subset		20 subset	
		Average $\uparrow$	Last $\uparrow$	Average $\uparrow$	Last $\uparrow$	Average $\uparrow$	Last $\uparrow$
EWC [18]	PNAS'17	19.01	6.00	15.82	3.79	12.35	4.73
EEIL [2]	ECCV'18	47.17	35.12	45.03	34.64	40.41	29.72
UCIR [12]	CVPR'19	50.30	39.42	48.58	37.29	42.84	30.85
MUC [23]	ECCV'20	32.23	19.20	26.67	15.33	21.89	10.32
PASS [45]	CVPR'21	49.54	41.64	47.19	39.27	42.01	32.93
GPM [29]	ICLR'21	6.10	4.72	5.95	4.54	5.69	4.28
DyTox [7]	CVPR'22	55.58	47.23	52.26	42.79	46.18	36.21
CLIP Zero-shot	ICML'21	69.62	65.30	69.55	65.59	69.49	65.30
Fine-tune	-	61.54	46.66	57.05	41.54	54.62	44.55
LwF [20]	TPAMI'17	60.97	48.77	57.60	44.00	54.79	42.26
iCaRL [28]	CVPR'17	77.02	70.39	73.48	65.97	69.65	64.68
LwF-VR [5]	Arxiv'22	77.56	70.89	74.12	67.05	69.94	63.89
ZSCL [42]	ICCV'23	80.27	73.57	78.61	71.62	77.18	68.30
MoE4Adapters [40]	CVPR'24	81.12	76.81	80.23	76.35	79.96	75.77
Adapter-based CLIP + GPM [29]	-	<u>82.82</u>	78.24	<u>81.84</u>	76.54	81.12	75.58
Adapter-based CLIP + TRGP [22]	-	81.91	<u>78.34</u>	81.69	<u>77.94</u>	<u>81.49</u>	<b>77.89</b>
MoE4Adapters [40] + DMNSP (Ours)	-	81.19	77.68	80.80	76.51	80.57	76.06
Adapter-based CLIP + DMNSP (Ours)	-	<b>83.28</b>	<b>79.14</b>	<b>82.70</b>	<b>77.94</b>	<b>81.96</b>	<u>77.10</u>

Table 1. Comparison of different methods on the TinyImageNet dataset in CIL settings with 100 base classes. We label the best and second-best methods with bold and underline styles based on “Average” and “Last” accuracy scores (%).

Method	Venue	10 subset		20 subset		50 subset	
		Average $\uparrow$	Last $\uparrow$	Average $\uparrow$	Last $\uparrow$	Average $\uparrow$	Last $\uparrow$
UCIR [12]	CVPR'19	58.66	43.39	58.17	40.63	56.86	37.09
Bic [38]	CVPR'19	68.80	53.54	66.48	47.02	62.09	41.04
PODNet [6]	ECCV'20	58.03	41.05	53.97	35.02	51.19	32.99
DER [39]	CVPR'21	74.64	64.35	73.98	62.55	72.05	59.76
GPM [29]	ICLR'21	41.76	23.26	29.91	13.89	19.27	7.48
DyTox+ [7]	CVPR'22	74.10	62.34	71.62	57.43	68.90	51.09
DNE [15]	CVPR'23	74.86	70.04	-	-	-	-
CLIP Zero-shot	ICML'21	74.47	65.92	75.20	65.74	75.67	65.94
Fine-tune	-	65.46	53.23	59.69	43.13	39.23	18.89
LwF [20]	TPAMI'17	65.86	48.04	60.64	40.56	47.69	32.90
iCaRL [28]	CVPR'17	79.35	70.97	73.32	64.55	71.28	59.07
LwF-VR [5]	Arxiv'22	78.81	70.75	74.54	63.54	71.02	59.45
ZSCL [42]	ICCV'23	82.15	73.65	80.39	69.58	79.92	67.36
MoE4Adapters [40]	CVPR'24	85.21	77.52	83.72	76.20	83.60	<b>75.24</b>
Adapter-based CLIP + GPM [29]	-	<u>86.73</u>	<u>79.14</u>	<u>85.01</u>	<u>76.80</u>	82.81	73.60
Adapter-based CLIP + TRGP [22]	-	85.69	77.82	84.80	76.14	<b>83.70</b>	73.67
MoE4Adapters [40] + DMNSP (Ours)	-	86.12	78.01	83.97	75.25	83.64	<u>74.87</u>
Adapter-based CLIP + DMNSP (Ours)	-	<b>87.59</b>	<b>79.94</b>	<b>85.29</b>	<b>76.96</b>	<u>83.66</u>	74.58

Table 2. Comparison of different methods on the CIFAR100 dataset in CIL settings. We label the best and second-best methods with bold and underline styles based on “Average” and “Last” accuracy scores (%).

seeds. More details can be found in the Section D of Supplementary Materials.

**Metrics.** In continual learning, we adopt the evaluation approach from [40], measuring accuracy throughout the learning process by calculating the average accuracy across all subsets (“Average”) and the accuracy of the final subset (“Last”). Furthermore, following [3], we quantify the extent of forgetting in continual learning using the Average Forgetting Rate across all subsets (“Average FR”) and the Forgetting Rate for the final subset (“Last FR”). Additionally, we conduct experimental explorations using Forward transfer (FWT) and Backward transfer (BWT). More details are provided in the Section B of the Supplementary Material.

## 4.2. Experimental Results

According to the study in [42], we present the results of methods that do not rely on CLIP (listed before “CLIP Zero-shot” in Tables 1 and 2) [2, 6, 7, 12, 15, 18, 23, 38, 39, 45] and methods based on CLIP (listed after “CLIP Zero-shot” in Tables 1 and 2) [5, 20, 28, 40, 42]. Additionally, using “Adapter-based CLIP” as the backbone, we compare the effects of combining it with different strategies. Specifically, we combine it with the traditional single-layer null space gradient projection method GPM [29] to form the “Adapter-based CLIP + GPM” method; combine it with the “trust region” gradient mapping strategy to form the “Adapter-based CLIP + TRGP” method; combine it with our method to form the “Adapter-based

Method	5 subset		10 subset		20 subset	
	Avg. ↓	Last ↓	Avg. ↓	Last ↓	Avg. ↓	Last ↓
Adapter-based CLIP	3.34	7.97	4.56	8.75	5.33	9.79
Adapter-based CLIP + GPM	3.30	7.98	4.50	8.69	5.35	9.27
MoE4Adapters + DMNSP (Ours)	2.26	<b>4.16</b>	3.12	4.88	<b>3.23</b>	<b>6.86</b>
Adapter-based CLIP + DMNSP (Ours)	<b>2.15</b>	4.84	<b>2.65</b>	<b>4.84</b>	3.65	7.11

Table 3. Comparison of our method with another three adapter-based methods on the TinyImageNet dataset in CIL settings with 100 base classes in terms of ‘‘Avg.’’ and ‘‘Last’’ forgetting rates (%). We label the best method with bold.

Method	10 subset		20 subset		50 subset	
	Avg. ↓	Last ↓	Avg. ↓	Last ↓	Avg. ↓	Last ↓
Adapter-based CLIP	7.14	11.39	7.43	12.61	9.34	14.52
Adapter-based CLIP + GPM	7.14	11.29	7.38	12.61	9.33	14.51
MoE4Adapters + DMNSP (Ours)	4.95	9.28	5.95	10.67	7.47	12.44
Adapter-based CLIP + DMNSP (Ours)	<b>4.36</b>	<b>7.48</b>	<b>4.73</b>	<b>9.52</b>	<b>7.39</b>	<b>12.42</b>

Table 4. Comparison of our method with other adapter-based methods on CIFAR100 in CIL settings in terms of ‘‘Avg.’’ and ‘‘Last’’ forgetting rates (%). We label the best method with bold.

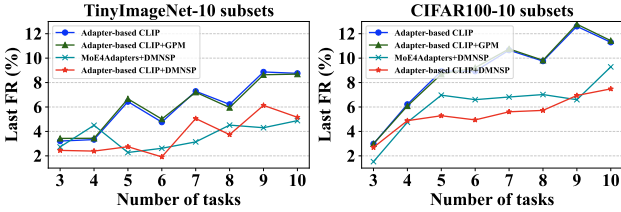


Figure 3. Comparison of forgetting curves between our method and another three adapter-based methods in CIL with 10 tasks.

CLIP + DMNSP’’ method; and merge DMNSP with the state-of-the-art method MoE4Adapters [40] to construct the ‘‘MoE4Adapters + DMNSP’’ method. GPM and TRGP are originally designed for task incremental learning; we have adapted them to CIL via classifier adjustments (see Supplementary Material for details).

**Accuracy results.** As shown in Tables 1 and 2, integrating various CL strategies with CLIP yields performance improvements of varying magnitudes. This observation suggests that leveraging the generalizable features inherent in VLM can significantly facilitate the model’s learning process in CL. Specifically, as depicted in Table 1, for all subset settings of TinyImageNet, our method attains the best results in terms of both average accuracy and last accuracy. Notably, as the number of tasks increases, compared with ‘‘Adapter-based CLIP + GPM’’ and the prior SOTA method ‘‘MoE4Adapters’’, our method shows increasingly prominent advantage in the last accuracy. As presented in Table 2, for the 10 and 20 subset settings of CIFAR100, although the superiority over other methods becomes less significant, our method can still achieve the overall best performance. For the 50 subset setting of CIFAR100, despite a significant increase in the number of tasks to be learned, DMNSP can still achieve the highest average accuracy. More results can be found in the Supplementary Material.

**Forgetting results.** We conduct further experimental exploration into the degree of forgetting. The original CLIP

Method	Train Params ↓	Memory Use ↓	Times ↓
LWF [20]	149.6M	32172MiB	1.54s/it
LWF-VR [5]	149.6M	32236MiB	1.51s/it
ZSCL [42]	149.6M	26290MiB	3.94s/it
MoE-Adapters [40]	59.8M	22358MiB	1.58s/it
Ours	<b>7.8M</b>	<b>21116MiB</b>	<b>0.23s/it</b>

Table 5. Comparison of computational cost during training on CIFAR100 between our method and others in terms of training parameters, GPU burden, and training time per iteration. All the experiments are completed on a single GeForce RTX 3090.

combined with the trainable layer-wise adapter strategy is taken as the ‘‘Adapter-based CLIP’’. Meanwhile, we also jointly compared the ‘‘Adapter-based CLIP + GPM’’ and ‘‘MoE4Adapters + DMNSP’’. As shown in Table 3 and Table 4, for the VLM, even when combined with the traditional null space gradient projection strategy, its anti-forgetting ability cannot be significantly improved. However, after being combined with our strategy, the last forgetting rate and the average forgetting rate of both MoE4Adapters and our method are significantly reduced, further demonstrating the effectiveness of our method in addressing the catastrophic forgetting issue.

We further present result of forgetting rate as depicted in Figure 3. Under the setting of CIL, starting from the third task, we record the forgetting rate of all previous tasks after completing each current task until all 10 tasks are accomplished. We observe that, for cases without the gradient projection strategy or with the traditional projection strategy, the forgetting rates are at relatively high levels. However, when combined with the DMNSP, the forgetting rates of the two methods are reduced to lower levels, indicating the effectiveness of our method. More results can be found in the Section C of the Supplementary Material.

**Complexity comparison.** We further compare the computational costs of our method with those of other methods to verify its parameter efficiency and time efficiency during training. Table 5 shows that methods described in [5, 20, 42] require full-tuning of all CLIP parameters. Consequently, these approaches demand extensive training resources, including a large number of parameters, substantial GPU usage, and considerable computational time. Although MoE4Adapters [40] only needs to fine-tune the adapter, due to its internal adoption of the Mixture of Experts structure [30] and routing selection mechanism, it also demands a relatively large number of training parameters and computing time. In contrast, despite using a simpler adapter structure, our approach reduces the number of training parameters by approximately 86.9% compared to MoE4Adapters, while also increasing training speed per iteration by roughly 6.8 times. This highlights the simplicity and efficiency of our method.

**Parameter variation and residual changes.** In the adapter-based VLM architecture, we conduct an in-depth

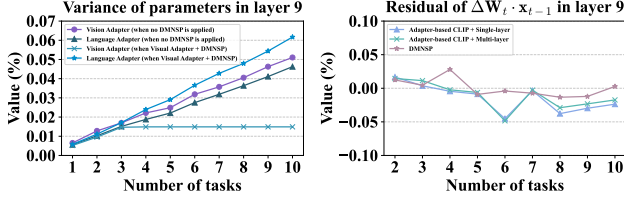


Figure 4. Parameter variation and residual changes in Adapter-based VLM before and after combination with DMNSP.

Method	Venue	10 subset		20 subset	
		Avg. ↑	Last ↑	Avg. ↑	Last ↑
L2P [37]	CVPR'22	80.83 $\pm$ 1.30	74.60 $\pm$ 0.90	78.39 $\pm$ 0.94	72.09 $\pm$ 1.12
DualPrompt [36]	ECCV'22	81.39 $\pm$ 1.25	74.87 $\pm$ 0.85	79.12 $\pm$ 1.27	71.69 $\pm$ 1.06
ESN [35]	AAAI'23	81.63 $\pm$ 1.10	75.11 $\pm$ 0.36	77.95 $\pm$ 0.76	70.57 $\pm$ 0.62
CODAprompt [31]	CVPR'23	81.32 $\pm$ 1.01	75.51 $\pm$ 0.81	78.07 $\pm$ 0.40	72.25 $\pm$ 0.78
LAE [10]	CVPR'23	76.71 $\pm$ 0.10	71.70 $\pm$ 0.39	73.72 $\pm$ 0.05	66.98 $\pm$ 0.35
InfLoRA [21]	CVPR'24	80.82 $\pm$ 0.24	75.65 $\pm$ 0.14	77.28 $\pm$ 0.45	71.01 $\pm$ 0.45
EASE [43]	CVPR'24	81.73	76.17	-	-
CPrompt [11]	CVPR'24	82.92 $\pm$ 0.70	77.14 $\pm$ 0.11	81.46 $\pm$ 0.93	74.79 $\pm$ 0.28
RAPF [16]	ECCV'24	85.85	79.62	86.28	79.62
VPT - NSP <sup>2</sup> [24]	NeurIPS'24	84.84 $\pm$ 0.12	78.88 $\pm$ 0.50	-	-
Adapter-based CLIP + GPM [29]	-	86.18 $\pm$ 0.08	80.11 $\pm$ 0.12	86.73 $\pm$ 0.19	80.65 $\pm$ 0.14
Adapter-based CLIP+ TRGP [22]	-	86.24 $\pm$ 0.10	81.29 $\pm$ 0.09	86.07 $\pm$ 0.17	81.21 $\pm$ 0.13
MoE4Adapters [40] + DMNSP (Ours)	-	86.45 $\pm$ 0.11	81.87 $\pm$ 0.16	87.06 $\pm$ 0.20	81.61 $\pm$ 0.18
Adapter-based CLIP + DMNSP (Ours)	-	87.49 $\pm$ 0.07	81.94 $\pm$ 0.15	86.81 $\pm$ 0.16	82.72 $\pm$ 0.09

Table 6. Comparison of Transformer-based methods on ImageNet-R under CIL with domain gap.

exploration of the variation patterns of adapter parameters and the numerical changes in the residual term  $\mathbf{W}_t \cdot \mathbf{x}_{t-1}$  (as described in Eq.(2)) of the model on old task data after learning new tasks.

As shown in Figure 4 (left), before the integration with the DMNSP, the variation amplitude of the parameters of the visual adapter (as shown by the purple line) is significantly higher than that of the language adapter (as shown by the green line). After the integration with DMNSP, the parameter updates of the visual adapter are projected into the null space. This mechanism effectively suppresses its parameter variance and enhances the stability of the model. In contrast, since the language adapter do not adopt the DMNSP strategy, during the joint update process, its variance further increases. This enables the language adapter to better adapt to the distribution of the new tasks, enhancing the plasticity of the model.

As can be seen from Figure 4 (right), as the null space gradient projection transitions from single-layer to multi-layer and finally the DMNSP is introduced, the residual term gradually approaches zero. This changing trend implies that the model’s resistance to forgetting is continuously enhanced. The above results strongly confirm the effectiveness of our processing of the visual modality.

**Superior cross-domain adaptation.** Given that the ImageNet-R dataset comprises complex cross-domain distribution shifts from natural images to artistic renditions [11] and includes hard samples that convolutional neural networks struggle to handle [10], we compare our approach with existing transformer-based methods that are adept at addressing domain gaps. The task setup is conducted according to [11]. As shown in Table 6, our method achieves superior performance. This indicates that our method main-

Components	Multi-layer	$\lambda$	Text	Vision	5 subset		10 subset		20 subset	
					Avg. ↑	Last ↑	Avg. ↑	Last ↑	Avg. ↑	Last ↑
Adapter-based CLIP + Single-layer	-	-	✓	✓	82.82 $\pm$ 0.02	78.24 $\pm$ 0.23	81.84 $\pm$ 0.31	76.54 $\pm$ 0.38	81.12 $\pm$ 0.27	75.58 $\pm$ 0.11
Adapter-based CLIP + Multi-layer	✓	-	✓	✓	82.77 $\pm$ 0.10	78.12 $\pm$ 0.15	81.84 $\pm$ 0.21	76.74 $\pm$ 0.14	81.16 $\pm$ 0.19	75.66 $\pm$ 0.15
DMNSP	✓	✓	✓	✓	82.99 $\pm$ 0.13	78.56 $\pm$ 0.25	81.79 $\pm$ 0.11	76.78 $\pm$ 0.09	81.28 $\pm$ 0.07	75.96 $\pm$ 0.14
DMNSP + $\lambda$	✓	✓	✓	✓	82.66 $\pm$ 0.09	77.22 $\pm$ 0.12	81.50 $\pm$ 0.05	75.06 $\pm$ 0.21	80.78 $\pm$ 0.16	75.58 $\pm$ 0.06
DMNSP + $\lambda$ + Multi-layer	✓	✓	✓	✓	82.83 $\pm$ 0.12	78.46 $\pm$ 0.2	81.79 $\pm$ 0.18	76.78 $\pm$ 0.13	81.11 $\pm$ 0.07	75.62 $\pm$ 0.13
DMNSP + $\lambda$ + Multi-layer + $\lambda$	✓	✓	✓	✓	83.28 $\pm$ 0.11	79.14 $\pm$ 0.16	82.70 $\pm$ 0.20	77.94 $\pm$ 0.13	81.96 $\pm$ 0.18	77.10 $\pm$ 0.15

Table 7. Ablation experiments on asymmetric adapters, multi-layer null spaces, and  $\lambda$  on TinyImageNet.

tains good adaptability in CL, even when confronted with complex data distributions.

### 4.3. Ablation Study

We conduct ablation studies to assess the effectiveness of asymmetric adapters and multi-layer null spaces in the proposed method. In Table 7, the first row presents the results of the traditional single-layer projection method applied to both modalities, which does not incorporate DMNSP or the dynamic projection coefficient  $\lambda$  from Eq.(6). When multi-layer null space projection is applied to both the visual and language modalities, the model exhibits limited plasticity due to overly restrictive gradient projection constraints. Consequently, combining it with  $\lambda$  does not yield significant performance improvements, as shown in rows 2 and 3. In contrast, since language parameters exhibit relatively low inherent variability, applying DMNSP solely to the language branch fails to effectively mitigate forgetting caused by changes in visual parameters (as shown in row 4). Similarly, applying single-layer gradient projection only to visual parameters, even when combined with  $\lambda$ , results in limited performance gains because of the insufficient constraints imposed on the residual term (as shown in row 5). Ultimately, the DMNSP strategy, which integrates multi-layer gradient projection with dynamic  $\lambda$ , achieves the highest performance. These results highlight the crucial role of the two components in reducing catastrophic forgetting and enhancing model performance. More experiment results are provided in the Section D of the Supplementary Material.

## 5. Conclusion

In this work, we propose a novel Dynamic Multi-layer Null Space Projection (DMNSP) strategy to tackle the catastrophic forgetting issue in Vision-Language Models (VLM) for continual learning. This strategy integrates a multi-layer null space gradient projection mechanism with a dynamic projection coefficient to realize a projection onto the common null subspace with minimal nonzero residuals. Moreover, by handling the visual and language branches of the VLM in an asymmetric manner, our method helps the VLM to achieve better continual learning ability. Extensive experiments have demonstrated that our method outperforms the state-of-the-art methods in terms of accuracy and resistance to forgetting. In the future, we plan to explore projection methods with more network architectures and modalities to further advance continual learning capabilities.

## Acknowledgement

This work is supported in part by the National Natural Science Foundation of China (62192783), Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), and the Australian Research Council’s Discovery Project (DP220101784).

## References

- [1] Ang Bian, Wei Li, Hangjie Yuan, Chengrong Yu, Zixiang Zhao, Mang Wang, Aojun Lu, and Tao Feng. Make continual learning stronger via c-flat. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1
- [2] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248, 2018. 6
- [3] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018. 6
- [4] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:16664–16678, 2022. 4
- [5] Yuxuan Ding, Lingqiao Liu, Chunna Tian, Jingyuan Yang, and Haoxuan Ding. Don’t stop learning: Towards continual learning for the clip model. *arXiv preprint arXiv:2207.09248*, 2022. 1, 2, 6, 7
- [6] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 86–102, 2020. 6
- [7] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9285–9295, 2022. 2, 6
- [8] Tao Feng, Mang Wang, and Hangjie Yuan. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2
- [9] Tao Feng, Wei Li, DiDi Zhu, Hangjie Yuan, Wendi Zheng, Dan Zhang, and Jie Tang. Zeroflow: Overcoming catastrophic forgetting is easier than you think. In *International Conference on Machine Learning (ICML)*, 2025. 1, 2
- [10] Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11483–11493, 2023. 8
- [11] Zhanxin Gao, Jun Cen, and Xiaobin Chang. Consistent prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28463–28473, 2024. 8
- [12] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 831–839, 2019. 2, 6
- [13] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning (ICLR)*, pages 2790–2799. PMLR, 2019. 1, 2
- [14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1
- [15] Zhiyuan Hu, Yunsheng Li, Jiancheng Lyu, Dashan Gao, and Nuno Vasconcelos. Dense network expansion for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11858–11867, 2023. 2, 6
- [16] Linlan Huang, Xusheng Cao, Haori Lu, and Xialei Liu. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 214–231. Springer, 2024. 8
- [17] Saurav Jha, Dong Gong, and Lina Yao. Clap4clip: Continual learning with probabilistic finetuning for vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 129146–129186. Curran Associates, Inc., 2024. 2
- [18] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 1, 2, 6
- [19] Wei Li, Tao Feng, Hangjie Yuan, Ang Bian, Guodong Du, Sixin Liang, Jianhong Gan, and Ziwei Liu. Unigrad-fs: Unified gradient projection with flatter sharpness for continual learning. *IEEE Transactions on Industrial Informatics*, 2024. 2
- [20] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 1, 6, 7
- [21] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23638–23647, 2024. 2, 5, 8
- [22] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Trgp: Trust region gradient projection for continual learning. In *International Conference on Learning Representations (ICLR)*, 2022. 3, 6, 8
- [23] Yu Liu, Sarah Parisot, Gregory Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–716, 2020. 6

- [24] Yue Lu, Shizhou Zhang, De Cheng, Yinghui Xing, Nannan Wang, PENG WANG, and Yanning Zhang. Visual prompt tuning in null space for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7878–7901. Curran Associates, Inc., 2024. 3, 5, 8
- [25] Zixuan Ni, Longhui Wei, Siliang Tang, Yueting Zhuang, and Qi Tian. Continual vision-language representation learning with off-diagonal information. In *International Conference on Machine Learning (ICML)*, pages 26129–26149. PMLR, 2023. 1
- [26] Jingyang Qiao, Zhizhong Zhang, Xin Tan, Yanyun Qu, Wensheng Zhang, and Yuan Xie. Gradient projection for parameter-efficient continual learning. *arXiv preprint arXiv:2405.13383*, 2024. 2
- [27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 1, 5
- [28] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017. 2, 6
- [29] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations (ICLR)*, 2021. 3, 4, 5, 6, 8
- [30] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 2, 7
- [31] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11909–11919, 2023. 8
- [32] Concetto Spampinato, Simone Palazzo, Isaak Kavasidis, Daniela Giordano, Nasim Souly, and Mubarak Shah. Deep learning human mind for automated visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6809–6817, 2017. 1
- [33] Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022. 1, 2
- [34] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben 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 (CVPR)*, pages 184–193, 2021. 2, 4
- [35] Yabin Wang, Zhiheng Ma, Zhiwu Huang, Yaowei Wang, Zhou Su, and Xiaopeng Hong. Isolation and impartial aggregation: A paradigm of incremental learning without interference. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 10209–10217, 2023. 8
- [36] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision (ECCV)*, pages 631–648. Springer, 2022. 8
- [37] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 139–149, 2022. 8
- [38] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 374–382, 2019. 2, 6
- [39] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3014–3023, 2021. 6
- [40] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. 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 (CVPR)*, pages 23219–23230, 2024. 2, 6, 7, 8
- [41] Zhen Zhao, Zhizhong Zhang, Xin Tan, Jun Liu, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Rethinking gradient projection continual learning: Stability/plasticity feature space decoupling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3718–3727, 2023. 2, 3, 4
- [42] Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xianguyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19125–19136, 2023. 1, 5, 6, 7
- [43] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23554–23564, 2024. 8
- [44] Da-Wei Zhou, Yuanhan Zhang, Yan Wang, Jingyi Ning, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Learning without forgetting for vision-language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2
- [45] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5871–5880, 2021. 6