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

Borui Kang [1], Lei Wang [2], Zhiping Wu [1], Tao Feng [3] Yawen Li [4], Yang Gao [1,5], Wenbin Li[1*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]University of Wollongong, Australia [3]Tsinghua University, China
[4]Beijing University of Posts and Telecommunications, China
[5]Yili Normal University, Xinjiang, China

A. Theoretical Framework of Multi-Layer Null Space Projection

Optimization objective for continual learning. Consider a deep neural network with L layers, where the parameters of the l-th layer are represented by $\mathbf{W}^l$, the input by $\mathbf{x}^{l-1}$, and the output is given by $\mathbf{x}^l = \mathbf{W}^l\mathbf{x}^{l-1}$. To prevent catastrophic forgetting when learning a new task, it is necessary to ensure that the output features for the old tasks $\mathcal{T}_{1:t-1}$ remain unchanged. Mathematically, this requires satisfying the following constraints:

$$\forall l \in \{1, \ldots, L\}, \quad (\mathbf{W}^l + \Delta\mathbf{W}^l)\mathbf{X}^{l-1} = \mathbf{W}^l\mathbf{X}^{l-1}, \quad (1)$$

where $\mathbf{X}^{l-1} = [\mathbf{x}_1^{l-1}, \ldots, \mathbf{x}_N^{l-1}]$ represents the matrix of output features from the $(l-1)$-th layer for the old tasks. Equivalently, the parameter updates must satisfy:

$$\Delta\mathbf{W}^l\mathbf{X}^{l-1} = 0 \quad \forall l. \quad (2)$$

This implies that the residual term should tend towards zero. The optimization problem can thus be formalized as:

$$\min_{\Delta\mathbf{W}} \mathcal{L}_{\text{new}}(\mathbf{W} + \Delta\mathbf{W}) \quad \text{s.t.} \quad \Delta\mathbf{W}^l\mathbf{X}^{l-1} = 0 \quad \forall l, \quad (3)$$

where $\mathcal{L}_{\text{new}}$ denotes the loss function for the new task.

Limitations of non-zero residuals. Single-layer projection methods focus solely on imposing constraints on each individual layer (e.g., the l-th layer) such that $\Delta\mathbf{W}^l\mathbf{X}^{l-1} = 0$, ignoring the influence of other layers. If the residual term for the current layer does not strictly tend towards zero, i.e., $\Delta\mathbf{W}^l\mathbf{X}^{l-1} \neq 0$, then $\mathbf{W}^l\mathbf{X}^{l-1} \neq \mathbf{W}^{l-1}\mathbf{X}^{l-1}$, and subsequent parameter updates propagate forward, gradually accumulating and ultimately causing the final output feature $\mathbf{x}^L$ to deviate from its original value, leading to catastrophic forgetting. The accumulated non-zero residuals $\Delta\mathbf{x}^L$ in the final output features directly degrade the network's performance on old tasks. For an old task input $\mathbf{x}^0$, the perturbed output becomes:

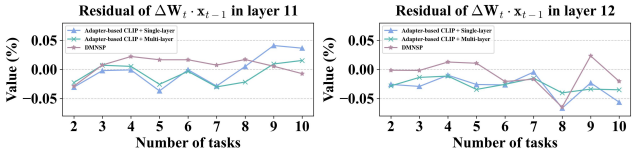


Figure 1. Residual comparisons of different layers on CIFAR100.

$$\mathbf{x}_{\text{new}}^L = \mathbf{x}^L + \Delta\mathbf{x}^L, \quad (4)$$

where $\mathbf{x}^L$ was originally mapped to the correct prediction (e.g., class label). The residual $\Delta\mathbf{x}^L$ disrupts this mapping, causing misclassification or regression errors. Mathematically, if the old task loss $\mathcal{L}_{\text{old}}$ is sensitive to $\mathbf{x}^L$ (e.g., cross-entropy loss), then:

$$\mathcal{L}_{\text{old}}(\mathbf{W}+\Delta\mathbf{W}) \approx \mathcal{L}_{\text{old}}(\mathbf{W})+\nabla_{\mathbf{x}^L}\mathcal{L}_{\text{old}}\cdot\Delta\mathbf{x}^L+\mathcal{O}(\|\Delta\mathbf{x}^L\|^2), \quad (5)$$

where the linear term $\nabla_{\mathbf{x}^L}\mathcal{L}_{\text{old}} \cdot \Delta\mathbf{x}^L$ dominates the forgetting. This is catastrophic forgetting in action: small output residuals induce large loss increases for old tasks.

Cumulative non-zero residuals also harm new task learning. Non-zero residuals alter the parameter manifold, steering updates away from the optimal subspace for the new task. For example, if the new task gradient $\nabla_{\mathbf{W}}\mathcal{L}_{\text{new}}$ is projected onto a perturbed null space (due to prior residuals), the effective update direction becomes suboptimal:

$$\Delta\mathbf{W}_{\text{eff}}^l = \mathbf{P}_{\text{Null}(\mathbf{X}^{l-1}+\Delta\mathbf{X}^{l-1})}\nabla_{\mathbf{W}^l}\mathcal{L}_{\text{new}}, \quad (6)$$

where $\mathbf{P}_{\text{Null}(\cdot)}$ is a corrupted projection matrix. This reduces learning efficiency.

Common Null Space. Define the null space of each layer as:

$$\text{Null}(\mathbf{X}^{l-1}) = \left\{\Delta\mathbf{W}^l \mid \Delta\mathbf{W}^l\mathbf{X}^{l-1} = 0\right\}. \quad (7)$$

*Corresponding Author

Multi-layer projection requires that the parameter updates simultaneously belong to the null spaces of all layers, i.e.,

$$\Delta \mathbf{W}^l \in \text{Null}(\mathbf{X}^{l-1}) \quad \forall l. \tag{8}$$

By projecting gradients onto each layer's null space sequentially, the final update direction becomes:

$$\Delta \mathbf{w} = \mathbf{P}^L \mathbf{P}^{L-1} \cdots \mathbf{P}^1 \mathbf{g}. \tag{9}$$

Theoretical derivation of multi-layer projection. If the objective of single-layer null space projection is $\Delta \mathbf{W}^l \mathbf{x}^{l-1} = 0$, then multi-layer null space projection automatically satisfies this goal, as it constrains the gradients to the common subspace of all layers' null spaces. Therefore, by further restricting the direction of parameter updates, multi-layer null space projection offers enhanced protection against catastrophic forgetting, ensuring that the residual terms more closely approach zero.

Experimental validation of residual reduction. To validate the effectiveness of multi-layer gradient projection in suppressing catastrophic forgetting, we conduct a layer-wise analysis of residual terms in the adapter-based CLIP framework. As depicted in Figure 1, during continual learning, we compared the residuals between the inputs of the old tasks and the parameters of the current task. Specifically, considering the semantic discriminativeness of the model's deep-layer features, we visualized the average residuals of the last two adapter layers, $\Delta \mathbf{W}_t \cdot \mathbf{x}_{t-1}$. The experimental results demonstrate that as the projection strategy evolves from single-layer projection to multi-layer projection, and further to DMNSP, the magnitude of the residuals gradually approaches zero. This validation underscores the necessity of multi-layer projection in visual-language continual learning and highlights the advantages of DMNSP over single-layer or static projection methods.

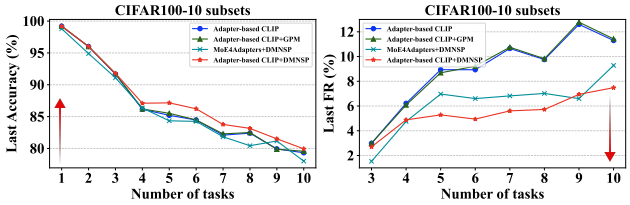
B. More Details about Metrics

Let $a_{t,j} \in [0,1]$ represent the accuracy evaluated on the held-out test set of the j-th task, with $j \leq t$, after incrementally training the network from tasks 1 to t. The average accuracy at task t is defined as

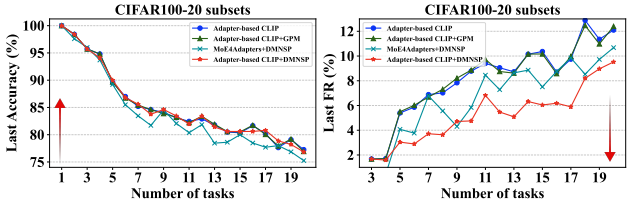
$$A_t = \frac{1}{t} \sum_{j=1}^{t} a_{t,j}. \tag{10}$$

Here, $a_{t,t}$ is denoted as "Last" accuracy and A_t is denoted as "Avg." accuracy. We quantify the forgetting of the j-th task after the model has been incrementally trained up to task t ($t > j$) as follows:

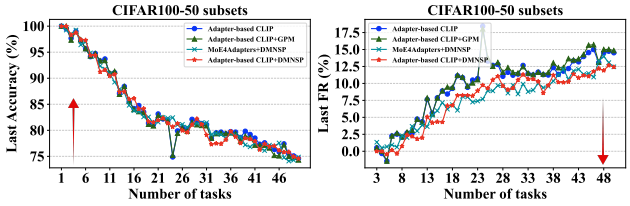
$$f_j^t = \max_{l \in \{1,\cdots,t-1\}} (a_{l,j} - a_{t,j}), \quad \forall j < t. \tag{11}$$



(a) Accuracy and forgetting curves on CIFAR100 with 10 tasks.



(b) Accuracy and forgetting curves on CIFAR100 with 20 tasks.



(c) Accuracy and forgetting curves on CIFAR100 with 50 tasks.

Figure 2. Comparison of accuracy and forgetting curves between our method and three other adapter-based approaches on the CIFAR100 CIL with various tasks.

Note that $f_j^t \in [-1,1]$ is defined for $j < t$. Moreover, by normalizing relative to the number of previously seen tasks, the average forgetting rate at the k-th task is written as

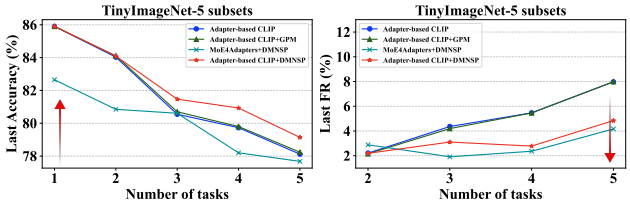
$$F_k = \frac{1}{k-1} \sum_{j=1}^{k-1} f_j^k. \tag{12}$$

f_t^t is denoted as "Last" forgetting rate and F_t is denoted as "Average" forgetting rate.

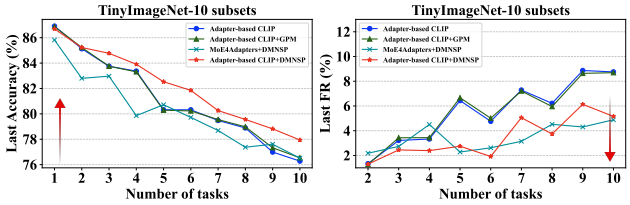
C. More Accuracy and Forgetting Curves

In Figures 2 and 3, we present comprehensive comparison graphs of the accuracy and forgetting rate curves under six CIL settings within the CIFAR100 and TinyImageNet datasets. It can be seen that the two methods incorporating DMNSP, namely "MoE4Adapters + DMNSP" and "Adapter-based CLIP + DMNSP", are relatively high in the accuracy curve and relatively low in the forgetting rate curve. This indicates that our methods have excellent plasticity and strong stability.

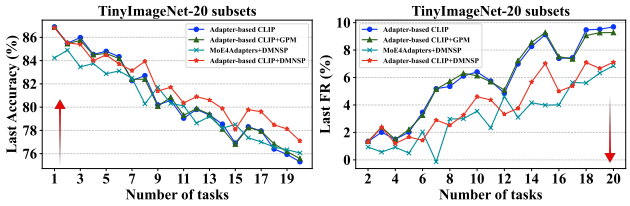
We introduce two new metrics, Forward transfer (FWT) and Backward transfer (BWT), to measure the ability of our method to transfer knowledge. Let $a_{t,j} \in [0,1]$ represent the accuracy evaluated on the held-out test set of the j-th



(a) Accuracy and forgetting curves on TinyImageNet with 5 tasks.

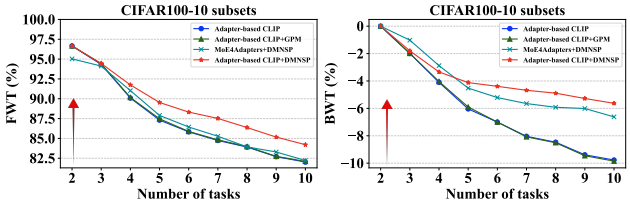


(b) Accuracy and forgetting curves on TinyImageNet with 10 tasks.

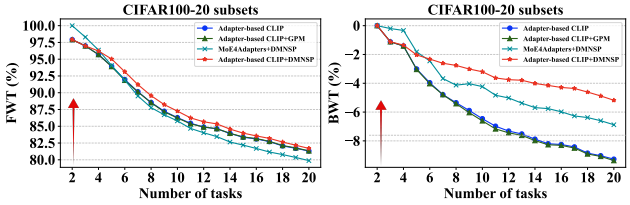


(c) Accuracy and forgetting curves on TinyImageNet with 20 tasks.

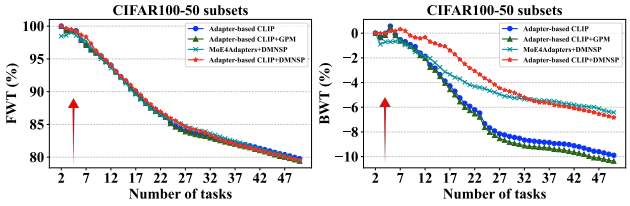
Figure 3. Comparison of accuracy and forgetting curves between our method and three other adapter-based approaches on the TinyImageNet CIL with various tasks.



(a) FWT and BWT curves on CIFAR100 with 10 tasks.



(b) FWT and BWT curves on CIFAR100 with 20 tasks.



(c) FWT and BWT curves on CIFAR100 with 50 tasks.

Figure 4. Comparison of FWT and BWT curves between our method and three other adapter-based approaches on the CIFAR100 CIL with various tasks.

task, with $j \leq t$, after incrementally training the network from tasks 1 to t. As shown below,

$$\text{FWT} = \frac{1}{t-1}\sum_{i=2}^{t}(a_{i-1,i} - a_{0,i}) \tag{13}$$

$$\text{BWT} = \frac{1}{t-1}\sum_{i=1}^{t-1}(a_{t,i} - a_{i,i}). \tag{14}$$

FWT represents the influence that learning a task t has on the performance on a future task. BWT represents the influence that learning a task t has on the performance on a previous task. As depicted in Figures 4 and 5, in most experiments, the methods incorporating DMNSP have demonstrated relatively superior FWT capability, indicating that the DMNSP strategy can provide good zero-shot learning potential. Additionally, the DMNSP-incorporated methods achieve high efficacy in BWT, reflecting the effectiveness of the DMNSP strategy in countering forgetting.

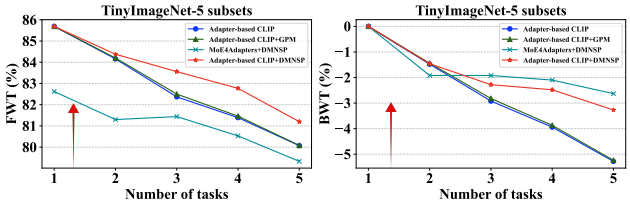
D. More Hyperparameter Analysis

Furthermore, as shown in Figure 6, we also explored the parameter ζ that controls the numerical range of the dynamic projection coefficient. Given that the gradient values diminish after each projection, it is necessary to moderately expand the numerical range. We observed that when ζ was set to 30, good performance could be achieved on both datasets. Based on this empirical observation, we fixed ζ at 30 and applied it uniformly across all experimental settings.

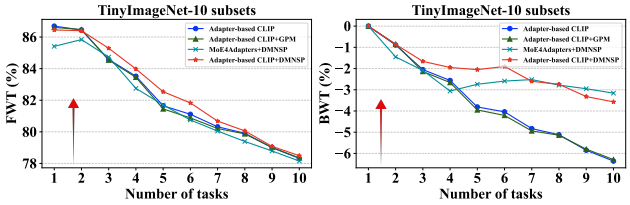
Additionally, we provide experimental insights into the selection of hyperparameters p and q. As illustrated in Table 1, given that our method focuses on approximating the null space and preserving the distinctiveness of null-space similarity across layers, we performed comparative experiments within the range of 0.5% to 10%. The experimental results indicate that optimal performance is achieved when both p and q are set to 1%. This demonstrates the effectiveness and robustness of our method under carefully selected hyperparameters.

E. More Experimental Results

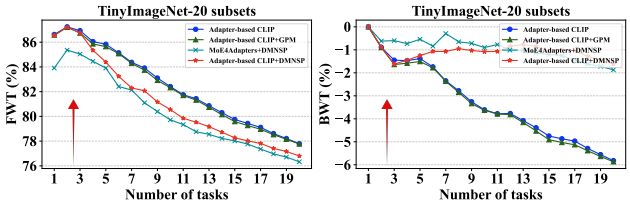
We use the maximum value of the logits output by the multi-classifier head as the method for GPM and TRGP to determine the task ID. Regarding the gradient memory update part, we have aligned both methods with our update strategy. On one hand, this can help both methods obtain more vector bases through updates; on the other hand, it can further align the memory consumption, thereby enabling a



(a) FWT and BWT curves on TinyImageNet with 5 tasks.



(b) FWT and BWT curves on TinyImageNet with 10 tasks.



(c) FWT and BWT curves on TinyImageNet with 20 tasks.

Figure 5. Comparison of FWT and BWT curves between our method and three other adapter-based approaches on the TinyImageNet CIL with various tasks.

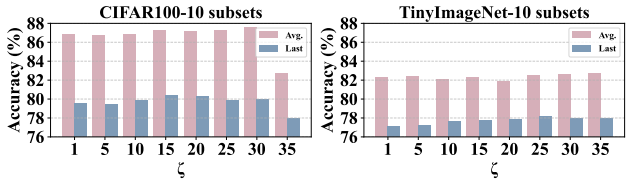


Figure 6. Impacts of ζ on the average accuracy and the last accuracy in CIL with 10 tasks.

Components		10 subset		20 subset		50 subset	
		Avg. ↑	Last ↑	Avg. ↑	Last ↑	Avg. ↑	Last ↑
$p = 1\%$	$q = 0.5\%$	$87.01_{\pm 0.1}$	$78.83_{\pm 0.16}$	$84.83_{\pm 0.08}$	$75.97_{\pm 0.05}$	$83.04_{\pm 0.11}$	$73.97_{\pm 0.16}$
	$q = 1\%$	$\mathbf{87.59}_{\pm 0.11}$	$\mathbf{79.94}_{\pm 0.16}$	$\mathbf{85.29}_{\pm 0.20}$	$\mathbf{76.96}_{\pm 0.13}$	$\mathbf{83.66}_{\pm 0.18}$	$\mathbf{74.58}_{\pm 0.15}$
	$q = 5\%$	$87.15_{\pm 0.23}$	$79.8_{\pm 0.11}$	$84.35_{\pm 0.06}$	$76.32_{\pm 0.2}$	$83.31_{\pm 0.02}$	$73.58_{\pm 0.13}$
	$q = 10\%$	$86.81_{\pm 0.11}$	$79.46_{\pm 0.2}$	$84.33_{\pm 0.19}$	$76.04_{\pm 0.12}$	$83.65_{\pm 0.07}$	$73.68_{\pm 0.11}$
$q = 1\%$	$p = 0.5\%$	$86.89_{\pm 0.14}$	$78.96_{\pm 0.09}$	$84.93_{\pm 0.11}$	$76.03_{\pm 0.15}$	$82.89_{\pm 0.16}$	$73.88_{\pm 0.21}$
	$p = 1\%$	$\mathbf{87.59}_{\pm 0.11}$	$\mathbf{79.94}_{\pm 0.16}$	$\mathbf{85.29}_{\pm 0.20}$	$\mathbf{76.96}_{\pm 0.13}$	$\mathbf{83.66}_{\pm 0.18}$	$\mathbf{74.58}_{\pm 0.15}$
	$p = 5\%$	$87.01_{\pm 0.07}$	$79.72_{\pm 0.2}$	$84.44_{\pm 0.19}$	$76.2_{\pm 0.07}$	$83.47_{\pm 0.14}$	$73.32_{\pm 0.12}$
	$p = 10\%$	$86.95_{\pm 0.2}$	$79.54_{\pm 0.14}$	$84.33_{\pm 0.22}$	$76.08_{\pm 0.15}$	$83.56_{\pm 0.11}$	$73.58_{\pm 0.18}$

Table 1. Hyperparameter selection comparison experiments. p and q represent the degree of approximation to the principal space and the degree of similarity measurement, respectively.

more fair comparison. We conducted more experiments on fine-grained datasets such as CUB (divided into 5 and 10 tasks) and StanfordCars (divided into 7 and 14 tasks). As shown in Table 2, the integration of the DMNSP strategy enabled the model to achieve superior performance, indicating the effectiveness of DMNSP for fine-grained datasets. Furthermore, we conducted additional comparisons with more vision-language-based methods. As shown in Table 3, DMNSP still achieves superior performance in 10-task CL benchmarks.

Method	Venue	CUB-200				StanfordCars			
		$T = 5$		$T = 10$		$T = 7$		$T = 14$	
		Last	Average	Last	Average	Last	Average	Last	Average
MoE4Adapters	CVPR'24	53.47	65.74	52.12	66.33	64.59	75.29	63.22	75.14
Adapter-based + GPM	-	57.44	70.64	52.74	67.99	68.96	81.49	65.33	77.14
Adapter-based + DMNSP (Ours)	-	**59.03**	**72.55**	**54.75**	**70.05**	**70.60**	**82.46**	**68.52**	**80.38**

Table 2. Comparison on fine-grained datasets under CIL.

Method	Venue	CIFAR100		ImageNet-R	
		Last	Average	Last	Average
MoE4Adapters	CVPR'24	77.52	85.21	65.77	72.80
CLAP4CLIP	NeurIPS'24	78.21	86.13	79.98	85.77
RAPF	ECCV'24	79.04	86.19	80.28	85.58
PROOF	TPAMI'25	79.05	86.70	80.10	85.32
Adapter-based + GPM	-	79.14	86.73	80.11	86.18
Adapter-based + DMNSP (Ours)	-	**79.94**	**87.59**	**81.94**	**87.49**

Table 3. Comparison with more multimodal methods under CIL.