PLAN: Proactive Low-Rank Allocation for Continual Learning

Xiequn Wang[*1]   Zhan Zhuang[*1,2]   Yu Zhang[†1]

[1] Southern University of Science and Technology   [2] City University of Hong Kong

{wangxiequn,yu.zhang.ust}@gmail.com, 12250063@mail.sustech.edu.cn

Abstract

Continual learning (CL) requires models to continuously adapt to new tasks without forgetting past knowledge. In this work, we propose Proactive Low-rank AllocatioN (PLAN), a framework that extends Low-Rank Adaptation (LoRA) to enable efficient and interference-aware fine-tuning of large pre-trained models in CL settings. PLAN proactively manages the allocation of task-specific subspaces by introducing orthogonal basis vectors for each task and optimizing them through a perturbation-based strategy that minimizes conflicts with previously learned parameters. Furthermore, PLAN incorporates a novel selection mechanism that identifies and assigns basis vectors with minimal sensitivity to interference, reducing the risk of degrading past knowledge while maintaining efficient adaptation to new tasks. Empirical results on standard CL benchmarks demonstrate that PLAN consistently outperforms existing methods, establishing a new state-of-the-art for continual learning with foundation models.

1. Introduction

Continual learning (CL) [22, 31, 33], also known as incremental learning or lifelong learning, is a learning paradigm in which a model processes and learns a sequence of tasks while preventing the catastrophic forgetting [5, 19] of previously acquired knowledge. CL plays a critical role in real-world applications such as autonomous driving [21, 25, 32] and robotics [15, 26], where models must continuously adapt to evolving and non-stationary environments. However, this setting inherently faces the stability–plasticity dilemma [11, 13, 40], requiring models to maintain stability on previously learned tasks while preserving sufficient plasticity to effectively acquire new information.

With the rise of large-scale pre-trained models, CL has found a strong foundation in freezing transferable representations, which helps preserve general knowledge and mit-
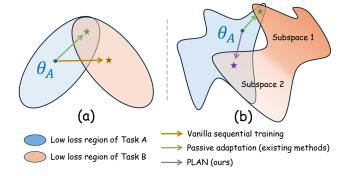

Figure 1. Conceptual illustration of PLAN compared to existing approaches. (a) Vanilla sequential training (orange arrow) causes parameter interference, moving the model away from previously low-loss regions. Existing methods (green arrow) passively avoid interference by enforcing orthogonality, typically assuming simplified low-loss regions for each task. (b) In practice, tasks possess multiple optimal regions. PLAN (purple arrow) is the first method to proactively optimize task-specific subspaces, explicitly anticipating future interference and robustly preserving performance within favorable regions for both current and previous tasks.

igate forgetting across tasks. Building on this, parameter-efficient fine-tuning (PEFT) techniques have emerged as promising solutions for CL [17, 27, 35–37]. By introducing only a small number of task-specific parameters, PEFT methods such as adapters [8], prompt tuning [16], and low-rank adaptation (LoRA) [9] enable new tasks to be learned efficiently with minimal disruption to existing knowledge. These lightweight modules as task vector [10, 39] incrementally encode task-specific updates while preserving the backbone's shared representations, making PEFT a natural fit for continual learning.

Among PEFT methods, LoRA [9] is highly effective, but applying it to CL is non-trivial. A simple strategy of allocating a separate LoRA module per task still faces a critical challenge: *how to ensure that updates for new tasks do not indirectly degrade the performance of previously learned tasks*?

Existing works like O-LoRA [35] and InfLoRA [17] ad-

---

dress this by enforcing orthogonality constraints, isolating new task updates from old ones. However, these methods adopt fundamentally **passive** strategies. Their primary goal is to prevent interference by avoiding shared subspaces, rather than actively identifying subspaces that are inherently more robust to future changes. This passive stance, while effective at reducing forgetting, limits the potential for more strategic adaptation.

To address this limitation, we propose the Proactive Low-rank AllocatioN a novel LoRA-based CL method that shifts from passive isolation to **proactive subspace planning**. Instead of merely preventing task conflicts, PLAN explicitly anticipates future interference during the training of each task and actively allocates subspaces to minimize potential conflicts across the entire task sequence. PLAN introduces two innovative mechanisms: **(1)** a perturbation-based optimization objective that anticipates worst-case interference scenarios during current task training, and **(2)** an orthogonal basis selection strategy informed by this objective, which proactively identifies optimal, low-interference directions in the parameter space for future tasks.

Specifically, each task-specific update in PLAN is represented via a low-rank decomposition ($\Delta W_t = B_t A_t$). PLAN first selects a fixed orthogonal basis for A_t from a predefined set, ensuring new tasks occupy distinct subspaces. Then, PLAN optimizes B_t using a novel min-max objective that robustly prepares the model against worst-case perturbations in the parameter directions reserved for future tasks. This forward-looking process not only makes the current task's parameters more robust but also guides the selection of the most stable basis vectors for the *next* task.

In summary, PLAN first advances LoRA-based continual learning through proactive interference mitigation, enabling robust lifelong adaptation of pre-trained models. Our contributions are summarized as follows.

- We shift LoRA-based CL from passive avoidance to proactive planning, where PLAN anticipates future conflicts and strategically selects task-specific subspaces.
- We introduce a min-max optimization strategy that anticipates worst-case perturbations, ensuring robust parameter updates and improved knowledge retention.
- We propose an efficient orthogonal basis selection mechanism that eliminates the need for additional subspace learning while maintaining a structured and interference-free representation space across tasks.
- Through extensive experiments, we show that PLAN surpasses existing CL methods across multiple benchmarks.

2. Related Work

Continual Learning with PEFT. With the advent of large pre-trained models, recent CL approaches increasingly adopt PEFT to mitigate forgetting. Among these, prompt-based methods such as L2P [37], DualPrompt [36],
and CODA-Prompt [27] introduce small trainable prompts for each task for ViTs [3]. However, these approaches face scalability challenges: as the number of tasks increases, the prompt pool grows linearly, requiring additional mechanisms to select appropriate prompts during inference. Moreover, with long task sequences, prompts tend to become homogeneous [6].

In contrast, LoRA-based methods [17, 35, 38, 39, 41] offer a more scalable alternative by introducing lightweight, task-specific weight updates. For example, O-LoRA [35] incrementally learns new tasks in subspaces orthogonal to all previous ones by constraining gradient updates to lie in the null space of past LoRA directions, thereby preventing interference without revisiting prior data. Similarly, InfLoRA [17] constructs interference-free subspaces by enforcing orthogonality constraints on low-rank adapters, ensuring that new task updates remain nearly orthogonal to those of earlier tasks. Both methods demonstrate the effectiveness of subspace isolation, showing that carefully selecting update directions can significantly reduce interference and achieve a stability–plasticity trade-off. Building on these insights, our work further explores subspace allocation, shifting from passive isolation toward more proactive strategies.

Min-Max Optimization in CL. The min-max objective has been explored to enhance robustness in CL. A notable example is FS-DGPM [2], which uses Sharpness-Aware Minimization (SAM) [4] to flatten the loss landscape of *past* tasks, making them more resilient to parameter changes. PLAN's objective is fundamentally different: instead of reacting to preserve past knowledge, it **proactively** perturbs the parameter space reserved for *future* tasks. This forward-looking optimization finds a solution for the *current* task that is already robust to anticipated future updates, and crucially, it informs the selection of the most stable subspaces for the *next* task.

Adaptive LoRA Techniques. The architecture of LoRA has been extensively refined for better efficiency and flexibility. Some works reduce parameter counts by sharing matrix components [24, 30, 42], while others enhance adaptability by decomposing weight updates into magnitude and direction [18, 41]. Initialization strategies have also been explored to better align with model properties or gradients [20, 34]. In contrast to these general-purpose improvements, PLAN introduces two innovations specifically for the continual learning challenge: 1) using a standard orthogonal basis for strict, interference-free subspace allocation, and 2) a proactive optimization strategy that enables forward-looking subspace planning.

3. Preliminaries

Problem Definition. Continual learning is formulated as the process of sequentially learning a series of tasks $\mathcal{T} =$
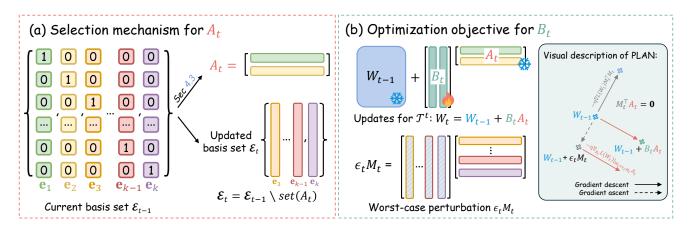
Figure 2. Overview of the proposed PLAN method. (a) Selection mechanism for A_t. For clarity, the case $t = 1$ is shown, where the current basis set $\mathcal{E}_0$ is the complete standard orthogonal basis. (b) Optimization objective for B_t using worst-case perturbations along unselected basis vectors, *i.e.* those not included in A_t, (denoted by $\mathcal{E}_t = set(M_t)$), with the gradient update directions visually indicated.

$\{\mathcal{T}^1, \mathcal{T}^2, \ldots, \mathcal{T}^N\}$, where each task $\mathcal{T}^t$ comprises a training dataset $\mathcal{D}_t = \{(\boldsymbol{x}_i^t, y_i^t)\}_{i=1}^{N_t}$ with N_t inputs $\{\boldsymbol{x}_i^t\}$ and the corresponding labels $\{y_i^t\}$. The primary objective of CL is to learn each task in sequence without incurring catastrophic forgetting of previously acquired knowledge.

Formally, for a model parameterized by θ, the goal of continual learning is to minimize the average generalization error over all encountered tasks:

$$\mathcal{L}_{CL}(\theta) = \frac{1}{j} \sum_{i=1}^{j} \mathcal{L}_i(\theta), \qquad (1)$$

where j denotes the index of the current task, and $\mathcal{L}_i(\theta)$ represents the generalization error of task $\mathcal{T}^i$. We use $L_S(\theta)$ to denote the empirical loss of the model on a dataset S. The model aims to perform well on both the current task $\mathcal{T}_j$ and all previously learned tasks. This paper focuses on class-incremental learning scenarios, using a pre-trained Vision Transformer (ViT) [3] as the initial classification model.

Low-Rank Adaptation. LoRA is a parameter-efficient fine-tuning technique for large pre-trained models. Given a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$ from the model, LoRA introduces two low-rank matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where the rank r is chosen to be much smaller than d and k. The weight update is then formulated as

$$W_{new} = W_0 + BA, \qquad (2)$$

with W_0 remaining fixed and only A and B being trainable.

4. Methodology

In this section, we introduce the proposed PLAN method.

4.1. Overview

Building upon LoRA, we train a dedicated LoRA adapter for each task while keeping the pre-trained weights and all

previously learned adapters fixed. For simplicity, we use a single MLP layer for illustration. Let W_0 denote the pre-trained weight and W_{t-1} the updated weight before learning from task $\mathcal{T}^t$. The weight update with the newly added LoRA for task $\mathcal{T}^t$ is then formulated as

$$W_t = \underbrace{W_0 + \sum_{i=1}^{t-1} B_i A_i}_{\text{frozen}} + B_t A_t = W_{t-1} + B_t A_t, \quad (3)$$

where $B_t \in \mathbb{R}^{d \times r_t}$ and $A_t \in \mathbb{R}^{r_t \times k}$, with r_t being the LoRA rank allocated for the new task. As illustrated in Figure 2, the proposed PLAN method incorporates two key designs to mitigate interference between previously learned and newly acquired knowledge: (1) constructing A_t by selecting a set of orthogonal basis vectors without additional training, and (2) optimizing B_t by applying perturbations along the directions of the unselected basis vectors.

Initialization. We begin by defining a standard basis set $\mathcal{E}_0 = \{\mathbf{e}_i\}_{i=1}^k$, where each k-dimensional vector $\mathbf{e}_i$ has all entries set to zero except for the i-th entry, which is set to 1. For each task $\mathcal{T}^t$, we select r_t basis vectors to form A_t from the current basis set and update the basis set as

$$\mathcal{E}_t = \mathcal{E}_{t-1} \setminus set(A_t), \qquad (4)$$

where $set(A_t)$ denotes the set of row vectors of A_t selected from $\mathcal{E}_{t-1}$, and the operation $\setminus$ indicates the removal of vectors from a set. For the first task, we construct A_1 by selecting the first r_1 basis vectors, *i.e.*, $A_1 = \mathbf{e}_{[1:r_1]}^\top$. We detail the selection strategy for A_t with rank r_t for subsequent tasks in Section 4.3 and describe the optimization of B_t for improved future allocation is described in Section 4.2.

In practice, the total number of basis vectors k corresponds to the feature dimension of the model, which is typically large and significantly greater than the number of tasks

encountered in continual learning scenarios. As a result, the basis set $\mathcal{E}_t$ rarely becomes empty in realistic settings, ensuring sufficient subspace capacity for long task sequences.

4.2. Optimization Objective for B_t

For task $\mathcal{T}^t$, once A_t is determined to allocate a task-specific subspace, we optimize the corresponding parameters in B_t to effectively capture task-specific information.

In this section, we introduce a min-max optimization objective that offers two key advantages. First, it proactively mitigates parameter conflicts across tasks by optimizing the current task's matrix B_t to be robust against interference from future tasks. Second, by explicitly analyzing perturbation sensitivity along different basis-vector directions, our method provides critical insights for selecting optimal initial basis vectors for subsequent tasks, a strategy we describe in detail in Section 4.3.

In the proposed PLAN method, basis vectors are orthogonal, meaning that $\mathbf{e}_i^\top \mathbf{e}_j = 0$ for $i \neq j$. Since each A_t is constructed from a unique set of these basis vectors, A_t is of full rank and remains orthogonal to A_i from other tasks (i.e., $A_i^\top A_j = 0$ for $i \neq j$). Consequently, the set $\{A_t\}_{t=1}^{M}$ projects the input onto distinct subspaces, allowing the model to learn new task features without interfering with previously acquired knowledge.

Let M_t denote the matrix formed by the row vectors corresponding to the unselected basis set $\mathcal{E}_t$:

$$M_t = \operatorname{span}(\mathcal{E}_t). \tag{5}$$

Since a future task $\mathcal{T}^s$ $(s > t)$ will select its A_s from the remaining rows of M_t, we introduce a min-max optimization for B_t to proactively reduce potential interference from future tasks. The objective is formulated as:

$$\min_{B_t} \max_{\|\epsilon_t\|_p \leq \rho} L_{\mathcal{D}_t}(W_{t-1} + B_t A_t + \epsilon_t M_t), \tag{6}$$

where ρ is a hyperparameter controlling the perturbation magnitude, $|M_t|$ is the number of basis vectors in M_t, $\epsilon_t \in \mathbb{R}^{d \times |M_t|}$ represents the worst-case perturbation applied to the unallocated subspace M_t, and $\| \cdot \|_p$ denotes the ℓ_p norm of a vector after transforming matrices to vectors. Here the term $\epsilon_t M_t$ models represents the possible interference by future tasks.

To solve problem (6), we approximate the inner maximization problem via the first-order Taylor expansion as

$$\arg \max_{\|\epsilon\|_p \leq \rho} L_{\mathcal{D}_t}(W_t + \epsilon M_t)$$
$$\approx \arg \max_{\|\epsilon\|_p \leq \rho} \left[L_{\mathcal{D}_t}(W_t) + \epsilon^\top \nabla_{W_t} L_{\mathcal{D}_t}(W_t) M_t^\top \right]$$
$$= \arg \max_{\|\epsilon\|_p \leq \rho} \epsilon^\top \nabla_{W_t} L_{\mathcal{D}_t}(W_t) M_t^\top. \tag{7}$$

Problem (7) has a closed-form solution as

$$\hat{\epsilon}_t(W_t) = \rho \frac{|\mathbf{g}|^{q-1} \odot \operatorname{sign}(\mathbf{g})}{(\|\mathbf{g}\|_q^q)^{\frac{1}{p}}}, \tag{8}$$

where $\mathbf{g} = \nabla_{W_t} L_{\mathcal{D}_t}(W_t) M_t^\top$, q satisfies $\frac{1}{p} + \frac{1}{q} = 1$, $|\mathbf{g}|$ denotes the elementwise absolute value of $\mathbf{g}$, the operator $\odot$ indicates elementwise multiplication, and $\operatorname{sign}(\mathbf{g})$ returns the elementwise sign of $\mathbf{g}$. By plugging $\hat{\epsilon}_t(W_t)$ into problem (6), the objective for B_t is formulated as

$$\min_{B_t} L_{\mathcal{D}_t}(W_{t-1} + B_t A_t + \hat{\epsilon}_t(W_t) M_t). \tag{9}$$

To avoid the computation of the Hessian matrix and reduce the computational cost, we treat $\hat{\epsilon}_t(W_t)$ as a constant instead of a function of W_t or B_t and write it as $\hat{\epsilon}_t$. Then we can solve problem (9) via stochastic gradient descent or its variants with the gradient computed as

$$\nabla_{B_t} L_{\mathcal{D}_t}(W_{t-1} + B_t A_t + \hat{\epsilon}_t(W_t) M_t)$$
$$\approx \nabla_{B_t} L_{\mathcal{D}_t}(W_t)\big|_{W_{t-1} + B_t A_t + \hat{\epsilon}_t M_t} A_t. \tag{10}$$

4.3. Selection Mechanism for A_{t+1}

After learning B_t as introduced in the previous section, we now introduce the selection mechanism for A_{t+1} in the proposed PLAN method. During the training of the previous task $\mathcal{T}^t$, we compute perturbations for each mini-batch using the formulation in Eq. (8). Specifically, to solve the inner maximization problem at each step s, we calculate the 2-norm of each column in the perturbation matrix $\hat{\epsilon}_t$ as:

$$n_{t,j}^s = \|\hat{\epsilon}_{t,j}\|_2, \tag{11}$$

where $\hat{\epsilon}_{t,j}$ denotes the j-th column of $\hat{\epsilon}_t$.

We then define a frequency function $h(i)$ that counts the number of times index i appears among these r smallest values over the last S training steps (i.e., a sliding window of size S):

$$h(i) = \sum_{s'=s-S+1}^{S} \mathbb{I}\left(i \in \arg \min_r n_{t,j}^{s'} \,\Big|\, j = 1, \dots, |M_t| \right), \tag{12}$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $\arg \min_r$ returns the set of r indices with the smallest values. The parameter S represents the size of the sliding window for accumulating frequencies. As shown in our analysis in Appendix A.1, a small value for S is sufficient, and we use $S = 50$ in our experiments.

These indices with the highest frequencies are considered most significant, as they consistently exhibit minimal perturbation. Consequently, for the subsequent task $\mathcal{T}^{t+1}$, we select the index set I_{t+1} containing the r indices with the highest frequency values to form A_{t+1} as

$$I_{t+1} = \arg \max_{I \subseteq \{1, \dots, |M_t|\}, |I| = r_{t+1}} \sum_{i \in I} h(i). \tag{13}$$

Method	ImageNet-R ($N = 5$)		ImageNet-R ($N = 10$)		ImageNet-R ($N = 20$)	
	Acc ↑	AAA ↑	Acc ↑	AAA ↑	Acc ↑	AAA ↑
L2P	64.20 (±0.30)	69.25 (±0.63)	62.52 (±0.41)	68.69 (±0.35)	58.63 (±0.52)	65.67 (±0.33)
Dual-Prompt	67.43 (±1.13)	71.40 (±0.85)	64.59 (±1.24)	69.59 (±0.72)	60.89 (±0.62)	66.20 (±0.51)
CODA-Prompt	74.52 (±4.25)	78.21 (±2.73)	71.58 (±0.26)	76.47 (±0.28)	67.10 (±0.46)	72.38 (±0.42)
Inc-LoRA	72.36 (±0.57)	79.60 (±0.27)	63.69 (±0.84)	74.54 (±0.35)	52.12 (±0.72)	67.73 (±0.45)
O-LoRA	73.12 (±6.09)	77.33 (±3.67)	65.74 (±0.81)	72.89 (±0.87)	59.94 (±0.82)	68.92 (±0.69)
InfLoRA	77.09 (±0.33)	**81.96** (±0.28)	74.37 (±0.54)	80.37 (±0.62)	69.83 (±0.65)	76.83 (±0.54)
PLAN (ours)	**77.79** (±0.24)	81.93 (±0.63)	**75.25** (±0.42)	**80.41** (±0.56)	**71.06** (±0.42)	**77.93** (±0.56)

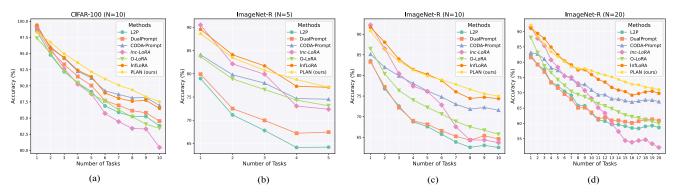Table 1. Comparison of different methods on *ImageNet-R* with varying N.

Figure 3. Variation of the performance of different methods during the learning of *ImageNet-R* and *CIFAR100*.

Algorithm 1 PLAN Method for Continual Learning

Input: a pre-trained ViT model f_θ, number of tasks T, training set $\{\{x_i^t, y_i^t\}_{i=1}^{n_t}\}_{t=1}^{T}$, number of training epochs E, predefined LoRA basis set $\mathcal{E}_0$.
Output: The learned LoRA parameters $\{A_t, B_t\}_{t=1}^{T}$.
for t in 1, ..., T **do**
  Construct A_t through Eqs. (12) and (13);
  $\mathcal{E}_t \leftarrow \mathcal{E}_{t-1} \setminus set(A_t)$;
  for $e = 1, ..., E$ **do**
    Sample batch $\mathcal{B} = \{(\boldsymbol{x}_1^t, y_1^t), ...(\boldsymbol{x}_b^t, y_b^t)\}$;
    $\mathbf{g} \leftarrow \nabla_{W_t} L_\mathcal{B}(W_t) M_t^\top$;
    Compute $\hat{\epsilon}_t(W_t)$ with $\mathbf{g}$ according to Eq. (7);
    $\mathbf{g}^{\text{PLAN}} \leftarrow \nabla_{B_t} L_\mathcal{B}(W_t)|_{W_{t-1}+A_t B_t + \hat{\epsilon}_t(W_t) M_t}$;
    Update B_t with $\mathbf{g}^{\text{PLAN}}$ through gradient descent;
  end for
end for

Finally, A_{t+1} is constructed by selecting the rows of M_t corresponding to the indices in I_{t+1}. This selection mechanism ensures that A_{t+1} comprises the basis vectors that consistently experience the least perturbation during training, thereby minimizing interference with previously acquired task knowledge. We summarize the whole process of our proposed PLAN method in Algorithm 1.

5. Experiments

This section presents the experimental setup and a comparison of the proposed PLAN method with other continual learning (CL) techniques across multiple benchmarks and foundation models.

5.1. Experimental Setup

Datasets. Following the approach in [17], we evaluate PLAN using three widely recognized CL benchmarks in the vision domain: CIFAR-100 [14], DomainNet [23], and ImageNet-R [7]. CIFAR-100 consists of 100 classes, ImageNet-R includes 200 ImageNet classes rendered in various artistic styles, and DomainNet features 345 classes spanning six distinct domains. For experimental purposes, we divide CIFAR-100 into 10-class subsets, ImageNet-R into tasks containing 40, 20, or 10 classes each (corresponding to 5, 10, or 20 tasks, respectively), and DomainNet into five tasks, each with 69 classes.

Evaluation Protocol. To evaluate the performance of continual learning, we employ two widely used metrics: **Average Accuracy (Acc)** and **Average Anytime Accuracy (AAA)**. Acc represents the mean classification accuracy across all tasks at the end of training, providing a final measure of overall performance. In contrast, AAA tracks the cu-

mulative average accuracy over all previously encountered tasks after training on each successive task, offering a more dynamic view of how well the model maintains knowledge throughout the learning process.

Baselines. The performance of PLAN is compared against several state-of-the-art CL methods, including L2P [37], DualPrompt [36], CODA-Prompt [27], O-LoRA [35], and InfLoRA [17]. Additionally, we introduce Incremental LoRA (*Inc-LoRA*) as a baseline to establish a lower bound for LoRA-based methods. *Inc-LoRA* involves training a separate LoRA for each new task and merging it into the original model after each task.

Architectural and Training Details. Following prior studies [17], we adopt the ViT-B/16 model, pre-trained on ImageNet-21K and fine-tuned on ImageNet-1K, as our backbone. Additionally, we evaluate our method on the self-supervised ViT-B/16 variant, iBOT [43], to assess its effectiveness across different training paradigms.

All models are trained using the Adam optimizer [12], which combines running averages of the gradient and squared gradient, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Training is conducted for 50 epochs on ImageNet-R, 20 epochs on CIFAR-100, and 5 epochs on DomainNet, with a batch size of 128. In line with [6], LoRA-based methods such as *Inc-LoRA*, O-LoRA [35], InfLoRA [17], and PLAN integrate LoRA modules into the key and value components of the attention mechanism. PLAN introduces a single hyperparameter, ρ, which is set to 0.01 across all datasets.

5.2. Main Results

Table 1 presents the performance of various methods on ImageNet-R across different task settings, while Table 2 shows the results on CIFAR-100 and DomainNet. Our proposed PLAN consistently outperforms previous methods. This improvement can be attributed to PLAN's proactive and orthogonality-based subspace allocation strategy, which effectively mitigates task interference by proactively selecting distinct parameter subspaces for each task.

In Figure 3, we further illustrate the evolution of accuracy across sequential tasks on ImageNet-R and CIFAR-100. Unlike previous approaches, which exhibit pronounced performance fluctuations and sharp accuracy drops when encountering new tasks, PLAN maintains more stable and higher accuracy levels throughout the learning process.

These empirical results demonstrate that proactively managing parameter update subspaces—rather than passively responding to interference after it occurs—enhances the stability and robustness of continual learning models.

5.3. Ablation Study

Ablation Study of PLAN Components. To validate the contributions of the two main components of PLAN, we conduct experiments to evaluate the effectiveness of our A_t selection mechanism and the optimization strategy for training B_t. In the first variant, we modify the A_t selection process to randomly select from $\mathcal{E}_t$. In the second variant, we retain the A_t selection mechanism but remove the B_t optimization algorithm, which is implemented by using the original PLAN method to construct $\mathcal{E}_t$, as the A_t selection mechanism depends on the B_t optimization algorithm. Finally, we remove both components, making the method equivalent to *Inc-LoRA*. Table 3 presents the results of this ablation study, which demonstrate that both components contribute to the effectiveness of PLAN. Both variants outperform *Inc-LoRA*, highlighting the importance of our basis construction for continual learning.

Variants of Basis Set Initialization. We evaluate three variants for constructing the basis set $\mathcal{E}$: random orthogonal basis, LoRA-GA initialization [34] and the standard orthogonal basis (ours).

For the random orthogonal basis, $\mathcal{E}$ is generated by sampling a matrix from a standard Gaussian distribution and then orthogonalizing it using the Gram-Schmidt process [29]. In LoRA-GA initialization, we first compute $\nabla_{W_0} L_{\{}(W_0)$ on the first task, where $L_{\{}$ is the full batch gradient over the first dataset, and then perform Singular Value Decomposition (SVD) to obtain $U, S, V \leftarrow \nabla_{W_0} L_{\{}(W_0)$. The initialization sets $A_1 \leftarrow V_{[1:r_1]}$ and the remaining basis as $\mathcal{E}_1 \leftarrow V_{[r+1:]}$.

The results in Table 6 show that although both random and LoRA-GA initialization methods ensure orthogonality, they perform worse than the standard orthogonal basis. In the case of random initialization, the basis vectors are orthogonal but misaligned with the input data, which limits the model's ability to leverage useful features and hinders learning performance.

In LoRA-GA, while the SVD operation effectively captures the dominant patterns from the first task and accelerates convergence, our experiments reveal a crucial drawback. Specifically, the LoRA selection mechanism tends to favor later singular vectors in the SVD decomposition. These later singular vectors are more likely to be nearly orthogonal to the input space relevant for future tasks. As a result, the new task's LoRA components, which are selected from these vectors, become poorly aligned with the input features required for subsequent tasks. This misalignment leads to significant difficulties in adapting to new tasks, reflecting the stability–plasticity dilemma: while the SVD-based initialization ensures high stability for the first task, it compromises the plasticity needed for learning future tasks, ultimately degrading performance on later tasks.

Method	CIFAR-100		DomainNet	
	Acc ↑	AAA ↑	Acc ↑	AAA ↑
L2P	83.81 (±0.42)	89.20 (±0.36)	70.26 (±0.25)	75.83 (±0.98)
Dual-Prompt	84.54 (±0.31)	90.02 (±0.22)	68.26 (±0.09)	73.84 (±0.45)
CODA-Prompt	86.95 (±0.36)	91.39 (±0.25)	70.58 (±0.53)	76.68 (±0.44)
Inc-LoRA	80.45 (±1.20)	88.40 (±0.48)	68.26 (±0.09)	73.84 (±0.45)
O-LoRA	83.41 (±0.46)	89.05 (±0.56)	70.58 (±0.53)	76.68 (±0.44)
InfLoRA	86.50 (±0.71)	91.23 (±0.38)	71.59 (±0.23)	**78.29** (±0.50)
PLAN (ours)	**87.54** (±0.31)	**92.21** (±0.35)	**72.12** (±0.16)	77.52 (±0.37)

Table 2. Comparison of different methods on *CIFAR-100* and *DomainNet* ($N = 5$).

Method	ImageNet-R ($N = 5$)		ImageNet-R ($N = 10$)		ImageNet-R ($N = 20$)	
	Acc ↑	AAA ↑	Acc ↑	AAA ↑	Acc ↑	AAA ↑
Inc-LoRA	72.36 (±0.57)	79.60 (±0.27)	63.69 (±0.84)	74.54 (±0.35)	52.12 (±0.72)	67.73 (±0.45)
w/o A_t selection	75.97 (±0.77)	80.69 (±0.49)	72.14 (±0.51)	78.56 (±0.56)	68.35 (±0.53)	76.33 (±1.12)
w/o perturbation	76.66 (±0.76)	80.38 (±0.51)	74.97 (±0.40)	79.57 (±0.30)	70.65 (±0.68)	76.42 (±0.76)
PLAN (ours)	**77.79** (±0.24)	**81.93** (±0.63)	**75.25** (±0.42)	**80.41** (±0.56)	**71.06** (±0.42)	**77.93** (±0.56)

Table 3. Ablation study of our method for two component.

In conclusion, while all three approaches enforce orthogonality, the standard orthogonal basis proves most effective, likely due to its natural alignment with the input space—potentially influenced by pretraining mechanisms such as dropout regularization [28]. Though not universally optimal, it effectively captures the intrinsic data structure, enabling more robust continual learning.

p	1	2	∞
Acc (%)	80.32	87.54	79.94

Table 4. Comparison of different p values on *CIFAR-100*.

Ablation Study on p. We also design an ablation study on p and q. By default, we set $p = 2$ to balance the contribution of potential weights in the future. To explore extreme cases, we set $p \to \infty$ and $p = 1$. When $p \to \infty$, Eq. (8) becomes:

$$\hat{\epsilon}_t(W_t) = \rho \, \text{sign}(\mathbf{g}), \qquad (14)$$

which eliminates the magnitude of the gradient and retains only its direction. When $p = 1$, Eq. (8) transforms to

$$\hat{\epsilon}(W_{ij}) = \begin{cases} \frac{1}{\mathbf{g}_{ij}}, & \text{if } \mathbf{g}_{ij} = \max(\mathbf{g}) \\ 0, & \text{otherwise} \end{cases} \qquad (15)$$

The results, shown in Table 4, demonstrate that both extreme configurations lead to unstable outcomes. From a numerical perspective, the magnitude of Eq. (8) is much smaller than in Eq. (14), resulting in $W \ll \hat{\epsilon}(W)$ during the normal training of deep neural networks. This reasoning is similarly applicable to Eq. (15), where $g_{ij} \ll 1$ implies that W and $W_{ij} \ll \hat{\epsilon}(W_{ij})M_{ij}$, where i and j refer to the row and column indices corresponding to the maximum absolute value of g. Since our PLAN method involves a min-max problem, an imbalance in difficulty between the minimization and maximization components can lead to training instability. When gradient magnitudes vary significantly, the optimization may fail to converge or even diverge. This issue, which is common in min-max optimization [1], arises when one component's perturbations are disproportionately large or small, disrupting stable training.

5.4. Analysis of Parameter and Storage Efficiency

We compare the trainable parameters and storage requirements of various CL methods. The results are shown in Table 7. For prompt-based methods such as L2P, Dual-Prompt, and CODA-Prompt, the learnable parameters are incorporated into the added prompts, and the corresponding keys must be stored. Notably, O-LoRA [35] requires storing the previous LoRA A block to compute the orthogonal loss, while InfLoRA [17] necessitates storing the gradient space. In contrast, our PLAN method only requires storing a negligible number of basis indices., which significantly reduces the storage requirements for both training and inference.

Method	CIFAR-100		ImageNet-20	
	Acc ↑	AAA ↑	Acc ↑	AAA ↑
L2P	47.42 (±1.12)	65.26 (±0.59)	73.71 (±0.27)	81.61 (±0.43)
Dual-Prompt	57.38 (±0.36)	65.26 (±0.45)	69.61 (±0.83)	76.92 (±1.19)
CODA-Prompt	59.57 (±0.33)	66.05 (±0.30)	**78.78** (±0.65)	**86.63** (±0.43)
Inc-LoRA	64.10 (±0.45)	72.33 (±0.29)	71.20 (±0.80)	82.47 (±0.26)
O-LoRA	53.26 (±0.59)	63.03 (±1.25)	72.75 (±1.55)	81.46 (±1.27)
InfLoRA	65.28 (±0.37)	74.11 (±0.47)	78.11 (±0.35)	86.47 (±0.17)
PLAN (ours)	**65.93** (±0.90)	**74.46** (±1.03)	78.39 (±0.49)	86.61 (±0.86)

Table 5. Comparison of different methods on *CIFAR-100* and *ImageNet-R* ($N = 20$) with iBOT-1k.

Method	CIFAR-100		ImageNet-20	
	Acc ↑	AAA ↑	Acc ↑	AAA ↑
Random Orthgonal Basis	81.21 (±0.45)	89.54 (±0.35)	69.42 (±0.48)	77.94(±0.87)
LoRA-GA Basis	84.30 (±0.23)	91.14 (±0.40)	69.40 (±0.40)	77.51 (±0.82)
Standard Basis (ours)	**87.54** (±0.31)	**92.21** (±0.35)	**71.06** (±0.42)	**77.93** (±0.56)

Table 6. Comparison of different basis set $\mathcal{E}$ initialization methods.

Method	EP (M)	SF (M)
L2P	1.85	0
DualPrompt	14.65	0
CODA-Prompt	5.57	0
Inc-LoRA	1.41	0
O-LoRA	1.41	13.36
InfLoRA	0.70	67.35
PLAN (ours)	0.70	0

Table 7. Comparison of methods by Expended Parameters (EP) and Stored Features (SF) in MB on *ImageNet-R* (N=20).

LoRA with forward-looking subspace allocation and robust training objective. Unlike existing approaches that passively enforce orthogonality to mitigate interference, PLAN anticipates future conflicts and proactively assigns task-specific subspaces, ensuring interference-free knowledge retention while simultaneously fostering adaptability through perturbation-aware optimization. By strategically preparing for future updates rather than simply reacting to past interference, PLAN provides a more effective solution to the stability–plasticity dilemma.

5.5. Analysis of Pre-trained Model

We conducted experiments using a ViT-B/16 model pretrained with iBOT [43]. All experimental settings, except for the choice of the pre-trained model, remain consistent with those outlined in Section 5.1. Table 5 presents the results of different methods on CIFAR-100 and ImageNet-R ($N = 20$). Upon comparing these results with those presented in Table 1, we observe that all methods utilizing self-supervised pre-trained models yield lower performance compared to their counterparts with supervised pretrained models. However, in this context, we find that PLAN either outperforms most methods or shows comparable performance, demonstrating its robustness even with self-supervised pre-training.

6. Conclusion

In this paper, we introduce PLAN (Proactive Low-Rank Allocation), a novel continual learning method that enhances

Limitations and Future Work. While PLAN shows strong performance, we identify several avenues for future research. First, its strict orthogonality, while excellent for preventing forgetting, does not explicitly promote positive backward transfer; exploring methods to selectively relax orthogonality could be beneficial. Second, our experiments primarily focused on ViT-based models; extending and evaluating PLAN on other architectures like ConvNets or different data modalities would be a valuable next step. Finally, while the standard basis proved effective, exploring adaptive basis generation for highly heterogeneous task sequences remains an interesting direction.

Acknowledgement

References

[1] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 7

[2] Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening sharpness for dynamic gradient projection memory benefits continual learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2021. Curran Associates Inc. 2

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 3

[4] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021. 2

[5] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, pages 128–135, 1999. 1

[6] 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*, pages 11449–11459, 2023. 2, 6

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

[8] 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 *Proceedings of the International Conference on Machine Learning*, pages 2790–2799, 2019. 1

[9] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 1

[10] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022. 1

[11] Dongwan Kim and Bohyung Han. On the stability-plasticity dilemma of class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20196–20204, 2023. 1

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6

[13] 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*, pages 3521–3526, 2017. 1

[14] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009. 5

[15] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020. 1

[16] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021. 1

[17] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024. 1, 2, 5, 6, 7

[18] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024. 2

[19] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 1

[20] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072, 2024. 2

[21] Yuan Meng, Zhenshan Bing, Xiangtong Yao, Kejia Chen, Kai Huang, Yang Gao, Fuchun Sun, and Alois Knoll. Preserving and combining knowledge in robotic lifelong reinforcement learning. *Nature Machine Intelligence*, pages 1–14, 2025. 1

[22] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, pages 54–71, 2019. 1

[23] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019. 5

[24] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6527–6536, 2024. 2

[25] Khadija Shaheen, Muhammad Abdullah Hanif, Osman Hasan, and Muhammad Shafique. Continual learning for real-world autonomous systems: Algorithms, challenges and

frameworks. *Journal of Intelligent & Robotic Systems*, 105 (1):9, 2022. 1

[26] Qi She, Fan Feng, Xinyue Hao, Qihan Yang, Chuanlin Lan, Vincenzo Lomonaco, Xuesong Shi, Zhengwei Wang, Yao Guo, Yimin Zhang, Fei Qiao, and Rosa H. M. Chan. OpenLORIS-Object: A robotic vision dataset and benchmark for lifelong deep learning. In *2020 International Conference on Robotics and Automation (ICRA)*, pages 4767–4773, 2020. 1

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

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 7

[29] Gilbert Strang. *Linear algebra and its applications*. Thomson, Brooks/Cole, Belmont, CA, 2006. 6

[30] Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng-Zhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37:9565–9584, 2024. 2

[31] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1

[32] Eli Verwimp, Kuo Yang, Sarah Parisot, Lanqing Hong, Steven McDonagh, Eduardo Pérez-Pellitero, Matthias De Lange, and Tinne Tuytelaars. Clad: A realistic continual learning benchmark for autonomous driving. *Neural Networks*, 161:659–669, 2023. 1

[33] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[34] Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. *Advances in Neural Information Processing Systems*, 37:54905–54931, 2024. 2, 6

[35] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, Singapore, 2023. Association for Computational Linguistics. 1, 2, 6, 7

[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*, pages 631–648. Springer, 2022. 2, 6

[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*, pages 139–149, 2022. 1, 2, 6

[38] Xiwen Wei, Guihong Li, and Radu Marculescu. Online-lora: Task-free online continual learning via low rank adaptation. *arXiv preprint arXiv:2411.05663*, 2024. 2

[39] Martin Wistuba, Prabhu Teja Sivaprasad, Lukas Balles, and Giovanni Zappella. Continual learning with low rank adaptation. *arXiv preprint arXiv:2311.17601*, 2023. 1, 2

[40] Guile Wu, Shaogang Gong, and Pan Li. Striking a balance between stability and plasticity for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1124–1133, 2021. 1

[41] Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-loRA: Scalable decoupled low-rank adaptation for class incremental learning. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[42] Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv preprint arXiv:2308.03303*, 2023. 2

[43] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022. 6, 8

A. Appendix

A.1. Analysis on Hyperparameter S

The parameter S in Eq. (12) denotes the size of the sliding window used to accumulate frequencies for basis selection. To analyze its impact, we computed the top-10 selected basis indices in the final layer for a task on CIFAR-100, varying S from 1 to 100. As shown in Figure 4, the set of selected indices stabilizes very quickly. The indices chosen with a small window (e.g., $S = 50$) are nearly identical to those chosen with a full window ($S = 100$, equivalent to all training steps). This indicates that a short-term memory of perturbation sensitivity is sufficient for robust basis selection, justifying our choice of $S = 50$ for efficiency.
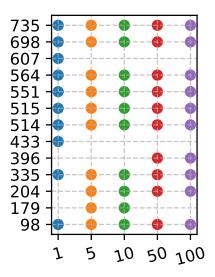

Figure 4. Top-10 selected basis indices (y-axis) for the next task as a function of the sliding window size S (x-axis). Each colored line tracks a specific basis index. The selection stabilizes with a small S.

A.2. Analysis on Hyperparameter ρ

The hyperparameter ρ controls the perturbation magnitude in our min-max objective (Eq. (8)). We performed an ablation study on *ImageNet-R* ($N = 5$) to determine its optimal value. The results are shown in Table 8. A value of $\rho = 0.01$ provides the best balance, leading to the highest performance. Larger values (e.g., 0.1) or smaller values (e.g., 0.001) resulted in slightly degraded performance, demonstrating the model's sensitivity to this parameter.

A.3. Discussion on Backward Transfer

Our work prioritizes stability, using strict orthogonal subspaces to effectively mitigate catastrophic forgetting, a success confirmed by our strong empirical results. This focus on interference prevention, however, means that PLAN does

Table 8. Ablation study on ρ on *ImageNet-R* ($N = 5$).

ρ	Acc (%)	AAA (%)
0.1	75.23	78.94
0.01	**77.79**	**81.93**
0.001	76.38	79.36

not explicitly facilitate positive backward transfer. Given that significant backward transfer is rarely observed in rehearsal-free CL, this represents a deliberate design choice. For future work, we believe exploring methods to dynamically adjust the degree of orthogonality could unlock opportunities for knowledge sharing across tasks while maintaining robustness.