Visual Prompt Tuning in Null Space for Continual Learning

Yue Lu[1], **Shizhou Zhang**[1]*, **De Cheng**[2]*, **Yinghui Xing**[1],
Nannan Wang[2], **Peng Wang**[1], **Yanning Zhang**[1]
[1] School of Computer Science, Northwestern Polytechnical University, China
[2] School of Telecommunications Engineering, Xidian University, China
zgxd@mail.nwpu.edu.cn, szzhang@nwpu.edu.cn, dcheng@xidian.edu.cn,
xyh_7491@nwpu.edu.cn, nnwang@xidian.edu.cn, peng.wang@nwpu.edu.cn,
ynzhang@nwpu.edu.cn

Abstract

Existing prompt-tuning methods have demonstrated impressive performances in continual learning (CL), by selecting and updating relevant prompts in the vision-transformer models. On the contrary, this paper aims to learn each task by tuning the prompts in the direction orthogonal to the subspace spanned by previous tasks' features, so as to ensure no interference on tasks that have been learned to overcome catastrophic forgetting in CL. However, different from the orthogonal projection in the traditional CNN architecture, the *prompt gradient orthogonal projection* in the ViT architecture shows completely different and greater challenges, $i.e.$, 1) the high-order and non-linear self-attention operation; 2) the drift of prompt distribution brought by the LayerNorm in the transformer block. Theoretically, we have finally deduced two consistency conditions to achieve the *prompt gradient orthogonal projection*, which provide a theoretical guarantee of eliminating interference on previously learned knowledge via the self-attention mechanism in visual prompt tuning. In practice, an effective null-space-based approximation solution has been proposed to implement the *prompt gradient orthogonal projection*. Extensive experimental results demonstrate the effectiveness of anti-forgetting on four class-incremental benchmarks with diverse pre-trained baseline models, and our approach achieves superior performances to state-of-the-art methods. Our code is available at https://github.com/zugexiaodui/VPTinNSforCL.

1 Introduction

Continual learning (CL) is crucial for AI models to adapt to the ever-changing environment by learning sequentially arrived data, where the *catastrophic forgetting* is the key challenge [21, 28]. Recently, prompt tuning-based continual learning methods [40, 32, 34, 44, 10, 22, 38, 46, 20, 12, 18] have been attracting increasing attention due to their impressive performances in the CL field. Existing prompt tuning-based works tackle the downstream continual learning problem by selecting and updating relevant prompts, which is encoded with full task-specific knowledge while exploiting the general knowledge of the pre-trained ViTs [40, 39].

On the contrary, this paper aims to learn each task by tuning the prompts in the direction orthogonal to the subspace spanned by previous tasks' features, so as to ensure no interference with tasks that have been learned to overcome *catastrophic forgetting* in CL. It is worth noting that forgetting can be theoretically resolved by gradient orthogonal projection methods [43, 31, 36, 45], which have

*Corresponding authors

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-0253

been extensively explored especially when adapting CNN models. Nevertheless, it remains a huge gap to introduce the orthogonal projection-based methods of CNNs to visual prompt tuning due to the following challenges: 1) the high-order and non-linear self-attention operation; 2) the drift of prompt distribution brought by the LayerNorm in the transformer block. For the linear operation in convolution or fully-connected layers, the output features of old tasks can remain unchanged by updating the weights in the orthogonal subspace of previous input features. While for self-attention, three linear transformations are employed on input tokens, followed by high-order and non-linear operations for the self-attention interaction of tokens. It makes the relationship between the update of prompts and the output image tokens much more complex, far exceeding mere linearity.

In this work, we theoretically deduced two consistency conditions to achieve the *prompt gradient orthogonal projection*, which provide a theoretical guarantee of eliminating interference on previously learned knowledge via the self-attention mechanism in visual prompt tuning. To be concrete, we firstly take the full self-attention and LayerNorm into consideration and derive a strict condition for eliminating the interference through a comprehensive analysis of the forward propagation of the ViT layer. Then we further propose to convert the condition of self-attention into its two sufficient conditions, which enables us to address the challenge of high order and nonlinearity. Thirdly, we propose a constraint of invariant prompt distribution that removes the obstacle to the final simplification of the conditions brought by the LayerNorm. The consistency conditions reveal that if the prompt update can be orthogonal to (1) the normalized previous input image tokens projected with the second-order qkv-transformation matrices of the pre-trained model, and (2) the activated attention map generated by image queries and prompt keys, the interference in visual prompt tuning can be eliminated theoretically.

In practice, based on the proposed consistency conditions, an effective null-space-based approximation solution [36] has been proposed to implement the *prompt gradient orthogonal projection*, while the invariant prompt distribution constraint is implemented by incorporating a loss function which penalizes the drifting of prompt distribution over sequential tasks. We validate our Null-Space Projection for Prompts (NSP^2) approach on extensive class-incremental benchmarks: 10- and 20-split CIFAR-100, 10-split ImageNet-R [39] and 10-split DomainNet [38], with the sequential fine-tuning VPT and CLIP models as baselines. Our approach brings 4%$\sim$10% improvements in accuracy, and reduces 9%$\sim$17% forgetting, which is superior to state-of-the-art methods.

Our contributions are summarized as follows: (1) We introduce the orthogonal projection into the visual prompt tuning for continual learning, which comprehensively considers the full operations of a transformer layer on the interference problem. (2) Two sufficient consistency conditions for the self-attention and an invariant prompt distribution constraint for LayerNorm are theoretically deduced, based on which an effective null-space-based approximation solution is introduced to implement the prompt gradient orthogonal projection for visual prompt tuning. (3) Extensive experimental results demonstrate the effectiveness of anti-forgetting on four class-incremental benchmarks with diverse pre-trained baseline models, and our approach achieves superior performances to state-of-the-art methods.

2 Related Work

Prompting-Based Approaches: Most of the prompting-based approaches adopt a two-stage framework [37, 39, 14, 15, 32, 42, 34, 35, 11, 18, 19]: querying a group of prompts for an individual sample and using them to prompt the pre-trained models. For example, L2P [40] first selects a group of prompts from a prompt pool and then feeds them into the ViT. CPrompt [11] proposes to mitigate the gap between training and testing stages to enhance prediction robustness and boost prompt selection accuracy. These approaches essentially focus on acquisition of task-specific prompts tailored to individual samples. There are also several one-stage methods [2, 22, 38, 44, 20] based on prompt tuning. (1) Slowly updating trainable parameters [10, 44]: *e.g.*, LAE [10] updates an offline expert with a large momentum to reduce the change of features. (2) Expandable backbones [46, 20]: *e.g.*, EASE [46] trains a distinct lightweight adapter module for each new task, and designs a semantic mapping to complement the drift of old class prototypes. (3) Enhancing classifiers rather than focusing on learning features [38, 22, 12]: *e.g.*, ESN [38] proposes an anchor-based classifier alignment approach based on energy-based models. As introduced above, these works still lack of a theoretical solution to the interference problem for visual prompt tuning. In our work, we conduct a deep analysis of this problem and provide a theoretical guidance on eliminating the interference.

Orthogonal Projection-Based Approaches: Orthogonal projection-based approaches [43, 4, 8, 31, 36, 17, 45] can theoretically eliminate the interference of new tasks on old tasks for linear layers. OWM [43] constructs a projector to find the direction orthogonal to the input space. GPM [31] first projects new gradients to the subspace important to the old tasks and then subtracts the projected components for updating parameters. Adam-NSCL [36] projects the parameter updates to the approximate null space of previous inputs. However, due to the different relationships between parameter updates and outputs in the linear operation and self-attention, the consistency condition used in CNNs is not directly applicable to the prompt tuning in ViTs. In our work, we derive the consistency conditions for the visual prompt tuning, enabling the application of orthogonal projection-based approaches to it, where the null-space projection [36] is adopted in our approach to get an approximate solution efficiently. We notice that a recently emerged work PGP [26] implements GPM [31] to prompt-based frameworks. However, it obtains the same conclusion as that of the linear operation under a simplified attention, which limits its application and performance as compared in the appendix D .

3 Preliminaries

Continual Learning: In the setting of continual learning, a network $f(\cdot|\mathbf{\Theta})$ with parameters $\mathbf{\Theta}$ is sequentially trained on a stream of disjoint tasks $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_T\}$, where task $\mathcal{T}_t$ is associated with paired data $\{(\mathcal{X}_t^{<i>}, y_t^{<i>})_{i=1}^{|\mathcal{T}_t|}\}$ of size $|\mathcal{T}_t|$. When a task $\mathcal{T}_t$ arrives, the model $f(\cdot|\mathbf{\Theta})$ would be trained for the current task, while the data from previous tasks is unreachable.

Forward Propagation of Visual Prompt Tuning in ViT Layers: We describe the forward propagation process of the ViT layer for visual prompt tuning, as illustrated in Figure 1 . Let $\mathbf{X} \in \mathbb{R}^{N\times D}$ and $\mathbf{P} \in \mathbb{R}^{M\times D}$ denote the N input image tokens of a sample (including the pre-trained class token if available) and M prompts, respectively, where D is the dimension of each token. In the ViT layer, only the prompts $\mathbf{P}$ are trainable parameters. The remaining parameters in LayerNorm, qkv-transformations and subsequent MLP introduced below are pre-trained and kept frozen. We use $\mathbf{Z} = [\mathbf{X}; \mathbf{P}] \in \mathbb{R}^{(N+M)\times D}$ to denote the concatenated input tokens. First, they undergo the LayerNorm [1] operation $\mathrm{LN}(\cdot)$:

$$\mathrm{LN}(\mathbf{Z}) = \frac{\mathbf{Z} - \boldsymbol{\mu}_{\mathbf{Z}}}{\boldsymbol{\sigma}_{\mathbf{Z}}} \odot \boldsymbol{\alpha} + \boldsymbol{\beta}, \tag{1}$$

where $\boldsymbol{\mu}_{\mathbf{Z}}, \boldsymbol{\sigma}_{\mathbf{Z}} \in \mathbb{R}^{N+M}$, $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^{D}$. The $\odot$ and division here denote the element-wise (Hadamard) product and division, respectively. Note that the vectors $\boldsymbol{\mu}_{\mathbf{Z}}$, $\boldsymbol{\sigma}_{\mathbf{Z}}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are broadcasted to match the matrices of dimensions $(N + M) \times D$, enabling them to carry out operations with $\mathbf{Z}$. Then the normalized tokens are fed into the qkv-transformations:

$$\mathbf{Q}_{\mathbf{Z}} = \mathrm{LN}(\mathbf{Z})\mathbf{W}_q + \boldsymbol{b}_q,\ \mathbf{K}_{\mathbf{Z}} = \mathrm{LN}(\mathbf{Z})\mathbf{W}_k + \boldsymbol{b}_k,\ \mathbf{V}_{\mathbf{Z}} = \mathrm{LN}(\mathbf{Z})\mathbf{W}_v + \boldsymbol{b}_v, \tag{2}$$

where $\mathbf{W}_{\{q,k,v\}} \in \mathbb{R}^{D\times D}$. The vector $\boldsymbol{b}_{\{q,k,v\}} \in \mathbb{R}^{D}$ is broadcasted to a matrix of dimensions $(N + M) \times D$ to facilitate the addition operation. Next is the self-attention:

$$\mathbf{F}_{\mathbf{Z}} = f_{\mathrm{SA}}(\mathbf{Z}) = \mathrm{softmax}(\frac{\mathbf{Q}_{\mathbf{X}}\mathbf{K}_{\mathbf{Z}}^{\top}}{\sqrt{D}})\mathbf{V}_{\mathbf{Z}}, \tag{3}$$

where $\mathbf{Q}_{\mathbf{X}}$ denotes the image tokens serving as queries. Eq. (3) can be expanded as Affinity, softmax (on rows) and Aggregation operations:

$$\begin{cases} \mathbf{A}_{\mathbf{Z}} = f_{\mathrm{aff}}(\mathbf{Q}_{\mathbf{X}}, \mathbf{K}_{\mathbf{Z}}) = \dfrac{\mathbf{Q}_{\mathbf{X}}\mathbf{K}_{\mathbf{Z}}^{\top}}{\sqrt{D}} = \dfrac{\mathbf{Q}_{\mathbf{X}}\left[\mathbf{K}_{\mathbf{X}}^{\top} \quad \mathbf{K}_{\mathbf{P}}^{\top}\right]}{\sqrt{D}} \in \mathbb{R}^{N\times(N+M)}, & (4) \\ \mathbf{S}_{\mathbf{Z}} = \mathrm{softmax}(\mathbf{A}_{\mathbf{Z}}) = \mathrm{softmax}(\left[\mathbf{A}_{\mathbf{X}} \in \mathbb{R}^{N\times N} \quad \mathbf{A}_{\mathbf{P}} \in \mathbb{R}^{N\times M}\right]) = \left[\mathbf{S}_{\mathbf{X}} \quad \mathbf{S}_{\mathbf{P}}\right], & (5) \\ \mathbf{F}_{\mathbf{Z}} = f_{\mathrm{agg}}(\mathbf{S}_{\mathbf{Z}}, \mathbf{V}_{\mathbf{Z}}) = \mathbf{S}_{\mathbf{Z}}\mathbf{V}_{\mathbf{Z}} = \left[\mathbf{S}_{\mathbf{X}} \quad \mathbf{S}_{\mathbf{P}}\right] \begin{bmatrix} \mathbf{V}_{\mathbf{X}} \\ \mathbf{V}_{\mathbf{P}} \end{bmatrix} \in \mathbb{R}^{N\times D}. & (6) \end{cases}$$

It is worth noting that the rows of the attention map where the prompts serve as queries (*i.e.*, $\mathbf{Q}_{\mathbf{P}}$) do not need to be computed, as formulated in Eq. (4) and illustrated in Figure 1 . The reason is that in VPT-Deep [13], the output prompts of this ViT layer will be replaced with new trainable prompts in the subsequent layer. Omitting $\mathbf{Q}_{\mathbf{P}}$ has no impact on the output image tokens of the ViT layer, as

https://doi.org/10.52202/079017-0253

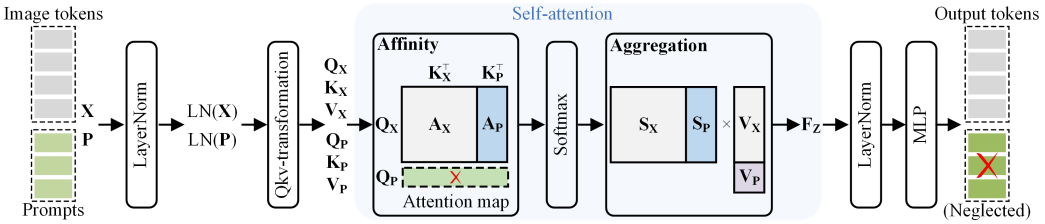

Figure 1: Illustration of the forward propagation in a ViT layer. Residual connections are omitted. The red crosses indicate the rows of attention map or the output prompts can be neglected.

the subsequent Aggregation, LayerNorm and MLP operations are performed independently for each token. If no new prompts are added in the next layer, the output prompts can be just discarded as well.

After the self-attention, operations consist of another LayerNorm and the MLP layer are applied individually to each token, without any interaction among the tokens. Finally, the output fine-tuned image tokens are fed into the next ViT layer.

Orthogonal Projection in Convolutional Layers: A convolutional operation is actually a linear operation. For a convolutional layer $f_{\text{conv}}(\cdot|\mathbf{\Theta}_t)$ in task $\mathcal{T}_t$, we use $\mathbf{\Theta}_t \in \mathbb{R}^{D_{\text{in}} \times D_{\text{out}}}$ to denote its unrolled convolutional kernel matrix [5]. Here, D_{in} represents the number of pixels within a kernel, and D_{out} corresponds to the number of kernels. Each convolutional patch from the input feature map is flattened into a row vector with a dimension of D_{in}. These row vectors of totaling n_p patches compose the input feature matrix $\mathbf{X}_t \in \mathbb{R}^{n_p \times D_{\text{in}}}$. The output feature for $\mathbf{X}_t$ in task $\mathcal{T}_t$ is expected to remain unchanged (referred to as consistent) in the next task $\mathcal{T}_{t+1}$ to prevent forgetting:

$$f_{\text{conv}}(\mathbf{X}_t|\mathbf{\Theta}_t) = f_{\text{conv}}(\mathbf{X}_t|\mathbf{\Theta}_{t+1}). \tag{7}$$

By substituting $\mathbf{\Theta}_{t+1} = \mathbf{\Theta}_t + \Delta\mathbf{\Theta}$, with $\Delta\mathbf{\Theta} \neq \mathbf{0}$ denoting the weight update in $\mathcal{T}_{t+1}$, the consistency condition for the convolutional layer is established as follows:

$$\mathbf{X}_t\mathbf{\Theta}_t = \mathbf{X}_t(\mathbf{\Theta}_t + \Delta\mathbf{\Theta}), \tag{8}$$

which can be further simplified as:

$$\mathbf{X}_t\Delta\mathbf{\Theta} = \mathbf{0}. \tag{9}$$

Eq. (9) suggests that if the weight update $\Delta\mathbf{\Theta}$ is orthogonal to the previous input feature $\mathbf{X}_t$ during training in the new task, the corresponding output feature will remain unchanged. Thereby, the interference of the new task on the old task is eliminated. This can be realized by projecting the candidate weight update $\mathbf{\Theta}_{\mathcal{G}}$ into the orthogonal subspace of $\mathbf{X}_t$: $\Delta\mathbf{\Theta} = \mathcal{P}\mathbf{\Theta}_{\mathcal{G}}$, where $\mathcal{P} \in \mathbb{R}^{D_{\text{in}} \times D_{\text{in}}}$ is an orthogonal projection matrix [43, 36, 31].

Similarly, for the prompt tuning which fine-tunes the prompts $\mathbf{P}_t$ in a ViT layer $f_{\text{vit}}(\mathbf{X}_t|\mathbf{P}_t)$, we also aim to satisfy the following consistency objective for the purpose of anti-forgetting:

$$f_{\text{vit}}(\mathbf{X}_t|\mathbf{P}_t) = f_{\text{vit}}(\mathbf{X}_t|\mathbf{P}_{t+1}). \tag{10}$$

However, the consistency condition in Eq. (9) does not hold for Eq. (10), since $f_{\text{vit}}(\mathbf{X}_t|\mathbf{P}_t) \neq \mathbf{X}_t\mathbf{P}_t$ in prompt tuning. Instead, all the tokens $\mathbf{X}_t$ and $\mathbf{P}_t$ first undergo a LayerNorm and then interact via the self-attention mechanism, as previously described. The complicated forward propagation within the ViT layer brings huge challenge to analyzing the consistency conditions in relation to the prompt update $\Delta\mathbf{P}$. In the next section, we will tackle this challenge and derive the consistency conditions for visual prompt tuning.

4 Method

We use $\mathbf{Z}_t = [\mathbf{X}_t; \mathbf{P}_t]$ and $\mathbf{Z}_{t+1} = [\mathbf{X}_t; \mathbf{P}_{t+1}]$ to denote the input tokens before and after updating the prompts, respectively, where $\mathbf{P}_{t+1} = \mathbf{P}_t + \Delta\mathbf{P}, \Delta\mathbf{P} \neq \mathbf{0}$. Our goal is to analyze how to satisfy Eq. (10) and derive one or more conditions expressed in terms of the prompt update $\Delta\mathbf{P}$. These conditions will subsequently guide the application of orthogonal projection to $\Delta\mathbf{P}$.

https://doi.org/10.52202/079017-0253

4.1 Analysis of Consistency Conditions

As can be seen in Figure 1, those outputs of LayerNorm and qkv-transformations corresponding to the image tokens remain unaffected by the updates to the prompts. Hence, the essence of attaining the consistency objective Eq. (10) can be turned into analyzing how to keep the output of self-attention in Eq. (3) unchanged as the prompts are updated, *i.e.*, satisfying:

$$\mathbf{F}_{\mathbf{Z}_t} = \mathbf{F}_{\mathbf{Z}_{t+1}}. \tag{11}$$

However, the nonlinear operation (*i.e.*, $\mathrm{softmax}$) and the potential higher-order term $\mathbf{W}_k^\top \mathbf{Z}^\top \mathbf{Z} \mathbf{W}_v$ arising from $\mathbf{K}_\mathbf{Z}^\top \mathbf{V}_\mathbf{Z}$ in Eq. (3) complicate the direct resolution of this objective. Specifically, the non-injection property of the $\mathrm{softmax}$ function causes non-unique solutions. The multiplication between $\mathbf{K}_{\mathbf{Z}_{t+1}}^\top \mathbf{V}_{\mathbf{Z}_{t+1}}$ derives a quadratic term $\mathrm{LN}(\mathbf{P}_t + \Delta\mathbf{P})^\top \mathrm{LN}(\mathbf{P}_t + \Delta\mathbf{P})$, which result in difficult optimization for $\Delta\mathbf{P}$.

To address this issue, we propose two sufficient conditions consisting solely of linear operations. Specifically, we split the process of self-attention into two primary stages, *i.e.*, the Affinity described by Eq. (4) and the Aggregation outlined in Eq. (6). We can achieve Eq. (11) by ensuring the consistency of each stage:

$$\begin{cases} f_{\text{aff}}(\mathbf{Q}_{\mathbf{X}_t}, \mathbf{K}_{\mathbf{Z}_t}) = f_{\text{aff}}(\mathbf{Q}_{\mathbf{X}_t}, \mathbf{K}_{\mathbf{Z}_{t+1}}), & (12) \\ f_{\text{agg}}(\mathbf{S}_{\mathbf{Z}_t}, \mathbf{V}_{\mathbf{Z}_t}) = f_{\text{agg}}(\mathbf{S}_{\mathbf{Z}_{t+1}}, \mathbf{V}_{\mathbf{Z}_{t+1}}). & (13) \end{cases}$$

We first analyze the consistency objective of Affinity, *i.e.*, Eq. (12), for $\mathbf{Z}_t$ and $\mathbf{Z}_{t+1}$:

$$\begin{cases} f_{\text{aff}}(\mathbf{Q}_{\mathbf{X}_t}, \mathbf{K}_{\mathbf{Z}_t}) = \mathbf{Q}_{\mathbf{X}_t} \left[\mathbf{K}_{\mathbf{X}_t}^\top \quad \mathbf{K}_{\mathbf{P}_t}^\top\right] = \left[\mathbf{Q}_{\mathbf{X}_t}\mathbf{K}_{\mathbf{X}_t}^\top \quad \mathbf{Q}_{\mathbf{X}_t}\left[\mathrm{LN}(\mathbf{P}_t)\mathbf{W}_k + \boldsymbol{b}_k\right]^\top\right], & (14) \\ f_{\text{aff}}(\mathbf{Q}_{\mathbf{X}_t}, \mathbf{K}_{\mathbf{Z}_{t+1}}) = \left[\mathbf{Q}_{\mathbf{X}_t}\mathbf{K}_{\mathbf{X}_t}^\top \quad \mathbf{Q}_{\mathbf{X}_t}\left[\mathrm{LN}(\mathbf{P}_{t+1})\mathbf{W}_k + \boldsymbol{b}_k\right]^\top\right], & (15) \end{cases}$$

where $\sqrt{D}$ is omitted for simplicity. Upon fulfilling Eq. (12), we can obtain $\mathbf{S}_{\mathbf{Z}_t} = \mathbf{S}_{\mathbf{Z}_{t+1}}$, corresponding to the output of Eq. (5). Subsequently, we analyze the consistency objective of Aggregation in Eq. (13), yielding results for $\mathbf{Z}_t$ and $\mathbf{Z}_{t+1}$ as:

$$\begin{cases} f_{\text{agg}}(\mathbf{S}_{\mathbf{Z}_t}, \mathbf{V}_{\mathbf{Z}_t}) = \mathbf{S}_{\mathbf{X}_t}\mathbf{V}_{\mathbf{X}_t} + \mathbf{S}_{\mathbf{P}_t}\mathbf{V}_{\mathbf{P}_t} = \mathbf{S}_{\mathbf{X}_t}\mathbf{V}_{\mathbf{X}_t} + \mathbf{S}_{\mathbf{P}_t}\left[\mathrm{LN}(\mathbf{P}_t)\mathbf{W}_v + \boldsymbol{b}_v\right], & (16) \\ f_{\text{agg}}(\mathbf{S}_{\mathbf{Z}_{t+1}}, \mathbf{V}_{\mathbf{Z}_{t+1}}) = f_{\text{agg}}(\mathbf{S}_{\mathbf{Z}_t}, \mathbf{V}_{\mathbf{Z}_{t+1}}) = \mathbf{S}_{\mathbf{X}_t}\mathbf{V}_{\mathbf{X}_t} + \mathbf{S}_{\mathbf{P}_t}\left[\mathrm{LN}(\mathbf{P}_{t+1})\mathbf{W}_v + \boldsymbol{b}_v\right]. & (17) \end{cases}$$

Based on Eq. (12−17), we are able to derive the following two equations, respectively:

$$\begin{cases} \mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top \mathrm{LN}(\mathbf{P}_t)^\top = \mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top \mathrm{LN}(\mathbf{P}_{t+1})^\top = \mathbf{Q}_{X_t}\mathbf{W}_k^\top \mathrm{LN}(\mathbf{P}_t + \Delta\mathbf{P})^\top, & (18) \\ \mathbf{S}_{\mathbf{P}_t}\mathrm{LN}(\mathbf{P}_t)\mathbf{W}_v = \mathbf{S}_{\mathbf{P}_t}\mathrm{LN}(\mathbf{P}_{t+1})\mathbf{W}_v = \mathbf{S}_{\mathbf{P}_t}\mathrm{LN}(\mathbf{P}_t + \Delta\mathbf{P})\mathbf{W}_v. & (19) \end{cases}$$

Note that we expect to further deduce Eq. (18) and Eq. (19) to obtain equations among $\mathrm{LN}(\mathbf{P}_t)$, $\mathrm{LN}(\mathbf{P}_t + \Delta\mathbf{P})$ and $\Delta\mathbf{P}$. However, due to the square root and quadratic terms in the expressions of the standard deviations $\boldsymbol{\sigma}_{\mathbf{P}_t}$ and $\boldsymbol{\sigma}_{\mathbf{P}_t+\Delta\mathbf{P}}$, it is difficult to express $\boldsymbol{\sigma}_{\mathbf{P}_t+\Delta\mathbf{P}}$ in terms of $\boldsymbol{\sigma}_{\mathbf{P}_t}$ and $\boldsymbol{\sigma}_{\Delta\mathbf{P}}$. Consequently, it is challenging to derive a straightforward equation that relates $\mathrm{LN}(\mathbf{P}_t)$ and $\mathrm{LN}(\mathbf{P}_t + \Delta\mathbf{P})$ through $\Delta\mathbf{P}$.

To simplify the problem, we introduce an additional constraint on the distribution of prompts. Concretely, we require that the updated prompts $\mathbf{P}_t + \Delta\mathbf{P}$ retain the same distribution as $\mathbf{P}_t$, *i.e.*, meeting the following assumption:

$$\begin{cases} \boldsymbol{\mu}_{\mathbf{P}_t+\Delta\mathbf{P}} = \boldsymbol{\mu}_{\mathbf{P}_t}, \\ \boldsymbol{\sigma}_{\mathbf{P}_t+\Delta\mathbf{P}} = \boldsymbol{\sigma}_{\mathbf{P}_t}. \end{cases} \tag{20}$$

In this way, we can establish a straightforward mathematical relationship connecting $\mathrm{LN}(\mathbf{P}_t + \Delta\mathbf{P})$, $\mathrm{LN}(\mathbf{P}_t)$ and $\Delta\mathbf{P}$:

$$\mathrm{LN}(\mathbf{P}_t+\Delta\mathbf{P}) = \frac{\mathbf{P}_t + \Delta\mathbf{P} - \boldsymbol{\mu}_{\mathbf{P}_t+\Delta\mathbf{P}}}{\boldsymbol{\sigma}_{\mathbf{P}_t+\Delta\mathbf{P}}} \odot \boldsymbol{\alpha} + \boldsymbol{\beta} = \frac{\mathbf{P}_t - \boldsymbol{\mu}_{\mathbf{P}_t} + \Delta\mathbf{P}}{\boldsymbol{\sigma}_{\mathbf{P}_t}} \odot \boldsymbol{\alpha} + \boldsymbol{\beta} = \mathrm{LN}(\mathbf{P}_t) + \frac{\Delta\mathbf{P}}{\boldsymbol{\sigma}_{\mathbf{P}_t}} \odot \boldsymbol{\alpha}. \tag{21}$$

https://doi.org/10.52202/079017-0253

Consequently, we can apply Eq. (21) to simplify Eq. (18) and (19) as:

$$\begin{cases} \mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top \mathrm{LN}(\mathbf{P}_t)^\top = \mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top \mathrm{LN}(\mathbf{P}_t)^\top + \mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top \Delta\mathbf{P}^\top / \boldsymbol{\sigma}_{\mathbf{P}_t}^\top \odot \boldsymbol{\alpha}^\top, & (22) \\ \mathbf{S}_{\mathbf{P}_t}\mathrm{LN}(\mathbf{P}_t)\mathbf{W}_v = \mathbf{S}_{\mathbf{P}_t}\mathrm{LN}(\mathbf{P}_t)\mathbf{W}_v + \mathbf{S}_{\mathbf{P}_t}\Delta\mathbf{P}\mathbf{W}_v / \boldsymbol{\sigma}_{\mathbf{P}_t} \odot \boldsymbol{\alpha}. & (23) \end{cases}$$

It should be noted that in Eq. 22 and Eq. 23 , $\mathbf{W}_k$, $\mathbf{W}_v$ and $\boldsymbol{\alpha}$ are pre-trained parameters kept frozen throughout the continual learning process. $\mathbf{Q}_{\mathbf{X}_t}$ and $\mathbf{S}_{\mathbf{P}_t}$ are two matrices derived from the input $\mathbf{X}_t$. As our objective is to ensure that the above two equations remain valid for the variables $\mathbf{Q}_{\mathbf{X}_t}$ and $\mathbf{S}_{\mathbf{P}_t}$, it is sufficient to meet the following conditions, in which $\mathbf{W}_v$ can be ignored whereas $\mathbf{W}_k$ remains crucial:

$$\begin{cases} \mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top \Delta\mathbf{P}^\top = \mathbf{0} & (24) \\ \mathbf{S}_{\mathbf{P}_t}\Delta\mathbf{P} = \mathbf{0} & (25) \end{cases}$$

Now we have obtained the simplified formulas expressed by $\Delta\mathbf{P}$ in Eq. (24) and (25).

To sum up, we convert the overall consistency equation Eq. (11) into two sufficient conditions Eq. (12) and (13) for Affinity and Aggregation, respectively. Consequently, we derive two corresponding consistency conditions Eq. (24) and (25) expressed by the prompt update $\Delta\mathbf{P}$, under the constraint of invariant prompt distribution formulated in Eq. (20). The deduced conditions can satisfy the consistency objective in Eq. (10), thereby achieving the goal of eliminating the interference of the new task on the old task for visual prompt tuning.

As $\mathbf{Q}_{\mathbf{X}_t} = \mathrm{LN}(\mathbf{X}_t)\mathbf{W}_q + \boldsymbol{b}_q$, Eq. (24) implies that if the (transposed) prompt update can be orthogonal to the normalized previous input image tokens $\mathbf{X}_t$ projected with a second-order transformation matrices $\mathbf{W}_q\mathbf{W}_k^\top$ of the pre-trained ViT, the consistency for Affinity can be guaranteed. When we ignore the normalization and the bias term in $\mathbf{Q}_{\mathbf{X}_t}$, Eq. (24) can be simplified as $\mathbf{X}_t\mathbf{W}_q\mathbf{W}_k^\top\Delta\mathbf{P}^\top = \mathbf{0}$. The simplified condition is still essentially different from the consistency condition of linear layers (*i.e.*, Eq. (9)) and that deduced in [26] (*i.e.*, $\mathbf{X}_t\Delta\mathbf{P}^\top = \mathbf{0}$). It indicates the interaction between the image tokens and prompts within ViT layers is fundamentally distinct, leading to a unique consistency condition related to the second-order transformation matrices $\mathbf{W}_q\mathbf{W}_k^\top$ of the pre-trained model. Moreover, Eq. (25) is also an essential condition served as one of the sufficient conditions for the consistency of the whole ViT layer. It implies that if the prompt update can be orthogonal to the activated attention map generated by the image queries ($\mathbf{Q}_\mathbf{X}$) and prompt keys ($\mathbf{K}_\mathbf{P}$), the consistency of Aggregation can be achieved.

4.2 Optimization of Consistency Conditions

To jointly optimize Eq. (24) and (25), we need to solve $\Delta\mathbf{P}$ that can meet both equations concurrently. Here, we employ a separate optimization approach to get an approximate solution efficiently. Initially, it ensures $\Delta\mathbf{P}^\top$ is orthogonal to the subspace spanned by $\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top$ to satisfy Eq. (24). Subsequently, it makes $\Delta\mathbf{P}$ orthogonal to the subspace spanned by $\mathbf{S}_{\mathbf{P}_t}$ to satisfy Eq. (25).

Specifically, we use $\mathbf{P}_\mathcal{G}$ to denote the candidate parameter update generated by the optimizer for the prompts. We aim to obtain a projection matrix $\mathcal{B}$ such that $\Delta\mathbf{P} = \mathcal{B}\mathbf{P}_\mathcal{G}$. Following the previously mentioned separate optimization strategy, we first ensure $\Delta\mathbf{P}^\top$ is orthogonal to $\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top$ by the projection matrix $\mathcal{B}_1$: $\Delta\mathbf{P}^\top = \mathcal{B}_1\mathbf{P}_\mathcal{G}^\top$. Then $\Delta\mathbf{P}$ is made orthogonal to $\mathbf{S}_{\mathbf{P}_t}$ by another projection matrix $\mathcal{B}_2$: $\Delta\mathbf{P} = \mathcal{B}_2\mathbf{P}_\mathcal{G}$. Therefore, the objective of the optimization turns into obtaining the two projection matrices $\mathcal{B}_1$ and $\mathcal{B}_2$ to satisfy Eq. (24) and (25). Inspired by the null-space projection method [36], the bases of $\mathcal{B}_1$ and $\mathcal{B}_2$ correspond to the null-space bases of $\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top$ and $\mathbf{S}_{\mathbf{P}_t}$, respectively. We use $\mathbf{U}_{1,0} \in \mathbb{R}^{D\times R_1}$ and $\mathbf{U}_{2,0} \times \mathbb{R}^{M\times R_2}$ to denote the bases of the null spaces for $\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top$ and $\mathbf{S}_{\mathbf{P}_t}$, where R_1 and R_2 indicate their nullities. $\mathbf{U}_{1,0}$ and $\mathbf{U}_{2,0}$ can be obtained from the right singular vectors associated with the zero singular values, through the process of singular value decomposition (SVD) applied by $\mathrm{SVD}((\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top)^\top\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top)$ and $\mathrm{SVD}(\mathbf{S}_{\mathbf{P}_t}^\top\mathbf{S}_{\mathbf{P}_t})$, respectively. In this way, we get the projection matrices $\mathcal{B}_1 = \mathbf{U}_{1,0}\mathbf{U}_{1,0}^\top \in \mathbb{R}^{D\times D}$ and $\mathcal{B}_2 = \mathbf{U}_{2,0}\mathbf{U}_{2,0}^\top \in \mathbb{R}^{M\times M}$, which are the solutions enabling $\Delta\mathbf{P}$ to jointly satisfy Eq. (24) and (25):

$$\Delta\mathbf{P} = \mathcal{B}_2\mathbf{P}_\mathcal{G}\mathcal{B}_1 = (\mathbf{U}_{2,0}\mathbf{U}_{2,0}^\top)\mathbf{P}_\mathcal{G}(\mathbf{U}_{1,0}\mathbf{U}_{1,0}^\top). \quad (26)$$

For the constraint Eq. (20), we incorporate an additional loss function aimed at penalizing the drift of prompt distribution, hence realizing a relaxed version of this constraint:

$$\mathcal{L}_{\mathrm{LN}} = \|\boldsymbol{\mu}_{\mathbf{P}_{t+1}} - \boldsymbol{\mu}_{\mathbf{P}_t}\|_1 + \|\boldsymbol{\sigma}_{\mathbf{P}_{t+1}} - \boldsymbol{\sigma}_{\mathbf{P}_t}\|_1. \quad (27)$$

Table 1: Comparison with the baselines ("-Seq") on four benchmarks using two types of models. The upper-bound means jointly training all the classes in the dataset.

Method	10S-CIFAR-100		20S-CIFAR-100		10S-ImageNet-R		10S-DomainNet	
	Acc. ↑	Forgetting ↓	Acc. ↑	Forgetting ↓	Acc. ↑	Forgetting ↓	Acc. ↑	Forgetting ↓
VPT-Seq	87.27	12.33	82.36	17.36	72.46	19.41	73.28	25.65
VPT-NSP^2	**91.74**	**3.28**	**89.89**	**4.91**	**78.88**	**5.06**	**83.54**	**8.54**
Upper-bound	93.87	-	93.87	-	84.60	-	89.25	-
CLIP-Seq	72.91	15.13	71.37	17.89	75.69	19.21	67.73	35.60
CLIP-NSP^2	**80.96**	**12.45**	**79.83**	**13.77**	**82.17**	**6.42**	**77.04**	**18.33**
Upper-bound	84.52	-	84.52	-	84.86	-	81.65	-

In Eq. (27), $\boldsymbol{\mu}_{\mathbf{P}_t}$ and $\boldsymbol{\sigma}_{\mathbf{P}_t}$ represent the target prompt distribution obtained in task $\mathcal{T}_t$, while $\boldsymbol{\mu}_{\mathbf{P}_{t+1}}$ and $\boldsymbol{\sigma}_{\mathbf{P}_{t+1}}$ denote the distribution to be optimized in task $\mathcal{T}_{t+1}$.

To sum up, we use Eq. (26) to realize Eq. (24) and (25), and use Eq. (27) to meet Eq. (20), thereby achieving the consistency objective Eq. (10) for anti-forgetting. We provide a full algorithm of our approach in the appendix A .

4.3 Extension to Multi-Heads

We further extend the consistency conditions Eq. (24) and (25) to multi-head self-attention, a common feature in current transformer-based models. Suppose there are H heads and $d = D/H$ represents the dimension of each token in a head. We use $\mathbf{Q}_{\mathbf{X}_t.h} \in \mathbb{R}^{N\times d}$, $\mathbf{W}_{k.h} \in \mathbb{R}^{D\times d}$ and $\mathbf{S}_{\mathbf{P}_t.h} \in \mathbb{R}^{N\times M}$ to denote the corresponding matrices in Eq. (24) and (25) for the h-th head, respectively. The objective is to ensure these conditions are met across all heads, *i.e.*, $\mathbf{Q}_{\mathbf{X}_t.h}\mathbf{W}_{k.h}^\top \Delta\mathbf{P}^\top = \mathbf{0}$ and $\mathbf{S}_{\mathbf{P}_t.h}\Delta\mathbf{P} = \mathbf{0}$, $\forall h \in \{1, 2, \cdots, H\}$. Let $\mathbf{\Omega}_{1,t} = [\mathbf{Q}_{\mathbf{X}_t.1}\mathbf{W}_{k.1}^\top; \cdots; \mathbf{Q}_{\mathbf{X}_t.H}\mathbf{W}_{k.H}^\top] \in \mathbb{R}^{HN\times D}$ and $\mathbf{\Omega}_{2,t} = [\mathbf{S}_{\mathbf{P}_t.1}; \cdots; \mathbf{S}_{\mathbf{P}_t.H}] \in \mathbb{R}^{HN\times M}$ represent the concatenated matrices from all the heads, respectively. Based on block matrix properties, those two sets of conditions can be formulated as $\mathbf{\Omega}_{1,t}\Delta\mathbf{P}^\top = \mathbf{0}$ and $\mathbf{\Omega}_{2,t}\Delta\mathbf{P} = \mathbf{0}$. To sum up, The main difference between single-head and multi-heads is that the parameter update should be orthogonal to the subspace spanned by the concatenation matrices from all heads for multi-heads self-attention. Therefore, for the multi-heads variant, only an additional step of concatenation of the matrices from all heads is required in our algorithm.

5 Experiments

5.1 Experimental Setups

In our experiments, we mainly utilize the VPT [13] with a ViT-B/16 backbone [9] pre-trained on ImageNet-21k. Additionally, we validate the effectiveness on the CLIP [27] model, wherein the visual prompts are inserted into the image encoder. Our experiments are conducted across 4 class-incremental benchmarks: 10- and 20-split CIFAR-100, 10-split ImageNet-R and 10-split DomainNet. We report the mean values of the final average accuracy and final average forgetting over 3 runs with different random seeds. Given that the null spaces of $\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top$ and $\mathbf{S}_{\mathbf{P}_t}$ may not always exist in practice, we compute the approximate null spaces and determine the nullities R_1 and R_2 in an adaptive manner, rather than the way suggested in [36]. For more detailed information regarding the experimental setups, please refer to Appendix B .

5.2 Main Results

Validation of Effectiveness: The comparison between our approach and the sequential fine-tuning VPT and CLIP baselines is shown in Table 1 . For the VPT model, the proposed NSP^2 achieves 4.47%∼10.26% improvements in accuracy on the 4 benchmarks. Meanwhile, it reduces the forgetting by 9.05%∼17.11%. As to the CLIP model, the NSP^2 improves the accuracy by 6.48%∼9.31%, and reduces the forgetting by 2.68%∼17.27%. We calculate the accuracy across all previously encountered tasks after completing training on each task. The accuracy curves of VPT-Seq and VPT-

https://doi.org/10.52202/079017-0253

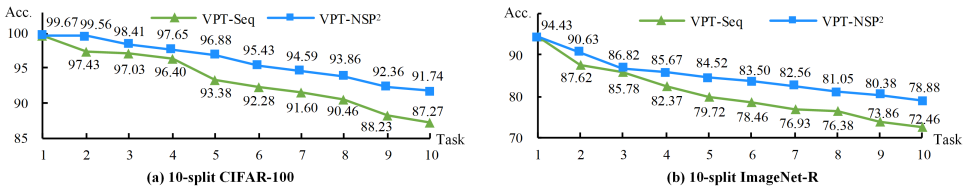

Figure 2: Task-by-task accuracy changing curves of VPT-Seq and VPT-NSP^2 on two benchmarks.

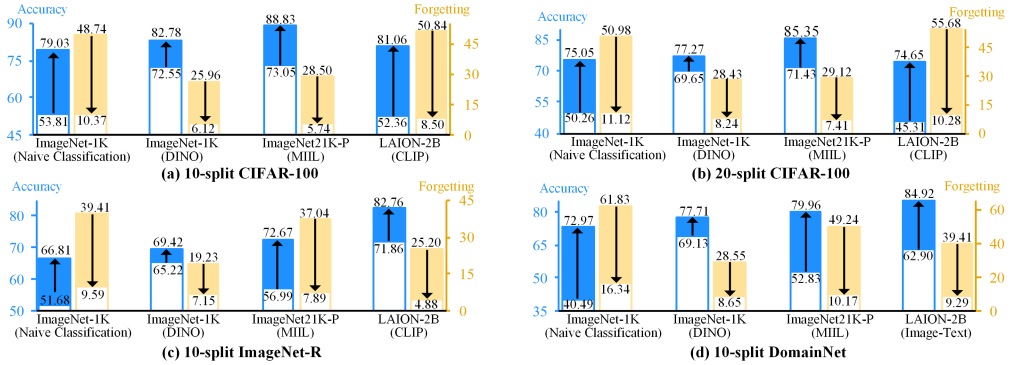

Figure 3: Results of utilizing different pre-training datasets and paradigms. The blue and yellow bars represent accuracy and forgetting, respectively. The upward arrows indicate the accuracy increasing from VPT-Seq to VPT-NSP^2, whereas the downward arrows denote the reduction in forgetting.

NSP^2 on 10-split CIFAR-100 and 10-split ImageNet-R are displayed in Figure 2. They demonstrate our approach consistently outperforms the baseline throughout the sequential learning of tasks.

We conduct additional experiments with the VPT model, utilizing the weights pre-trained on different datasets as well as different paradigms, as shown in Figure 3 . The pre-training paradigms and datasets include: naive classification on ImageNet-1k [30], DINO [3] on ImageNet-1k, MIIL [29] on ImageNet21k-P and CLIP on LAION-2B [6] (we only use its image encoder). As can be seen from the figure, our approach not only significantly enhances accuracy but also markedly mitigates forgetting. These results further demonstrate the generalizability of the proposed approach.

Comparison with Existing Methods: We compare our method with existing methods in Table 2, where the competitors include many recent works. The proposed VPT-NSP^2 achieves state-of-the-art performance on the four benchmarks, with surpassing the second best approach by an average of 1.49% in accuracy. The forgetting of our approach is not the lowest, which is reasonable since our approach sacrifices some stability for a better trade-off between stability and plasticity. The outperforming accuracy can demonstrate the superiority of our method.

Ablation Study: The two consistency conditions Eq. (24) and (25), along with the constraint Eq. (20), constitute the main components of our approach. They correspond to $\mathcal{B}_1$, $\mathcal{B}_2$ in Eq. (26), and $\mathcal{L}_{\text{LN}}$ in Eq. (27). We study their effects on the four benchmarks using VPT-NSP^2, with results presented in Table 3. We can see that the projection for Affinity ($\mathcal{B}_1$) plays a crucial role, which brings 3.31%$\sim$9.03% improvement in accuracy and 5.42%$\sim$14.76% decline in forgetting. Furthermore, the projection for Aggregation ($\mathcal{B}_2$) and the loss $\mathcal{L}_{\text{LN}}$ for invariant prompt distribution are indispensable as well for minimizing forgetting. Optimal accuracy is achieved when all three conditions are applied.

Model Analysis: We analyze the evolution of training losses on the 10-split CIFAR-100 and 10-split ImageNet-R benchmarks, as shown in Figure 4 . Each point on the curve represents the training loss of the data in $\mathcal{T}_1/\mathcal{T}_2$ after the model has been trained on subsequent tasks. As can be seen, the losses of VPT-NSP^2 on previous tasks can be almost retained, confirming that our approach can effectively mitigate the interference of new tasks on old tasks.

Trade-off between Stability and Plasticity: We first adaptively determine the nullities R_1 and R_2 for $\mathcal{B}_1$ and $\mathcal{B}_2$ to achieve near-minimum forgetting. Based on this, we assign two weights η_1 and η_2 to the projection matrices to control the trade-off between stability and plasticity: $\Delta\mathbf{P} =$

Table 2: Comparison with existing methods that use the pre-trained ViT-B/16 on ImageNet-21k. The standard deviations are also reported if available. Missing results in the corresponding papers are denoted as "-". The results marked with † and ‡ are implemented by [11] and [10], respectively. The highest accuracies are in bold, and the second highest accuracies are underlined.

Method	Venue	10S-CIFAR-100		20S-CIFAR-100		10S-ImageNet-R		10S-DomainNet	
		Acc.	Forgetting	Acc.	Forgetting	Acc.	Forgetting	Acc.	Forgetting
L2P [40]	CVPR'22	$83.83_{\pm 0.04}$	$7.63_{\pm 0.30}$	$80.10_{\pm 0.72}^{\ddagger}$	-	$61.57_{\pm 0.66}$	$9.73_{\pm 0.47}$	$81.17_{\pm 0.83}^{\dagger}$	$8.98_{\pm 1.25}$
DualPrompt [39]	ECCV'22	$86.51_{\pm 0.33}$	$5.16_{\pm 0.09}$	$82.02_{\pm 0.32}^{\ddagger}$	-	$68.13_{\pm 0.49}$	$4.68_{\pm 0.20}$	$81.70_{\pm 0.78}^{\dagger}$	$8.04_{\pm 0.31}$
CODA-P [32]	CVPR'23	$86.25_{\pm 0.74}$	$1.67_{\pm 0.26}$	-	-	$75.45_{\pm 0.56}$	$1.64_{\pm 0.10}$	$80.04_{\pm 0.79}^{\dagger}$	$10.16_{\pm 0.35}$
ESN [38]	AAAI'23	$86.34_{\pm 0.52}$	$4.76_{\pm 0.14}$	$80.56_{\pm 0.94}^{\ddagger}$	-	$62.61_{\pm 0.96}^{\ddagger}$	-	$79.22_{\pm 2.04}^{\dagger}$	$10.62_{\pm 2.12}$
APG [33]	ICCV'23	89.35	6.01	88.64	6.51	73.27	8.59	-	-
LAE [10]	ICCV'23	$85.59_{\pm 0.46}$	-	$83.93_{\pm 0.28}$	-	$72.66_{\pm 0.63}$	-	-	-
DualP-LGCL [15]	ICCV'23	$87.23_{\pm 0.21}$	$5.10_{\pm 0.15}$	-	-	$69.46_{\pm 0.04}$	$4.20_{\pm 0.06}$	-	-
C-LN [23]	ICCVW'23	$86.95_{\pm 0.37}$	$6.98_{\pm 0.43}$	-	-	$76.36_{\pm 0.51}$	$8.31_{\pm 1.28}$	-	-
EvoPrompt [18]	AAAI'24	$87.97_{\pm 0.30}$	$2.60_{\pm 0.42}$	$84.64_{\pm 0.14}$	$3.98_{\pm 0.24}$	$76.83_{\pm 0.08}$	$2.78_{\pm 0.06}$	$79.50_{\pm 0.29}$	$3.81_{\pm 0.36}$
OVOR-Deep [12]	ICLR'24	$85.99_{\pm 0.89}$	$6.42_{\pm 2.03}$	$84.13_{\pm 0.75}$	$6.81_{\pm 0.77}$	$76.11_{\pm 0.21}$	$7.16_{\pm 0.34}$	$79.61_{\pm 0.86}$	$4.77_{\pm 0.94}$
DualP-PGP [26]	ICLR'24	$86.92_{\pm 0.05}$	$5.35_{\pm 0.19}$	$83.74_{\pm 0.01}$	$7.91_{\pm 0.15}$	$69.34_{\pm 0.05}$	$4.53_{\pm 0.04}$	$80.41_{\pm 0.25}$	$8.39_{\pm 0.18}$
InfLoRA [20]	CVPR'24	$87.06_{\pm 0.25}$	$6.22_{\pm 0.39}$	$81.42_{\pm 0.54}$	$6.42_{\pm 0.33}$	$75.65_{\pm 0.14}$	$5.73_{\pm 0.44}$	$81.45_{\pm 0.68}$	$5.35_{\pm 0.52}$
EASE [46]	CVPR'24	87.76	5.94	85.80	7.19	76.17	7.82	78.89	7.89
CPrompt [11]	CVPR'24	$87.82_{\pm 0.21}$	$5.06_{\pm 0.50}$	$83.97_{\pm 0.31}$	$6.85_{\pm 0.43}$	$77.14_{\pm 0.11}$	$5.97_{\pm 0.68}$	$82.97_{\pm 0.34}$	$7.45_{\pm 0.93}$
VPT-NSP^2	This work	$\mathbf{91.74}_{\pm 0.63}$	$3.28_{\pm 0.45}$	$\mathbf{89.89}_{\pm 0.72}$	$4.91_{\pm 0.59}$	$\mathbf{78.88}_{\pm 0.50}$	$5.06_{\pm 0.26}$	$\mathbf{83.54}_{\pm 0.77}$	$8.54_{\pm 0.48}$

Table 3: Ablation studies of each component in our approach on the four benchmarks.

$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{L}_{\text{LN}}$	10S-CIFAR-100		20S-CIFAR-100		10S-ImageNet-R		10S-DomainNet	
			Acc. ↑	Forgetting ↓	Acc. ↑	Forgetting ↓	Acc. ↑	Forgetting ↓	Acc. ↑	Forgetting ↓
			87.27	12.33	82.36	17.36	72.46	19.41	73.28	25.65
√			90.58	6.91	88.13	10.27	78.05	8.14	82.31	10.89
	√		88.74	10.85	83.32	16.48	74.71	14.69	78.87	17.81
√	√		91.33	4.22	88.96	6.42	78.37	6.25	83.17	8.95
√		√	91.42	3.94	88.46	8.64	78.30	6.31	83.13	9.32
	√	√	89.36	9.32	86.67	11.59	75.27	13.35	79.45	16.50
√	√	√	**91.74**	**3.28**	**89.89**	**4.91**	**78.88**	**5.06**	**83.54**	**8.54**

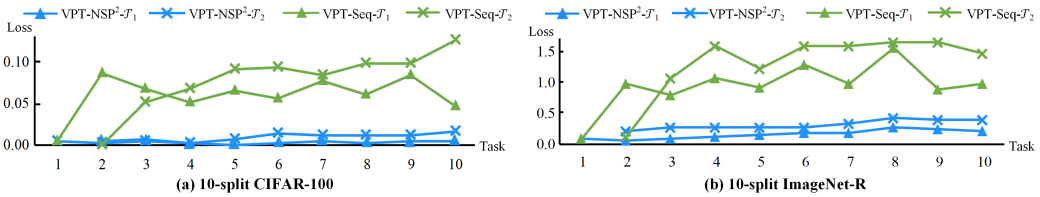

Figure 4: Training loss curves of VPT-NSP^2 and VPT-Seq on tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ when the models are trained on sequential tasks.

$[\eta_2\mathcal{B}_2 + (1-\eta_2)\mathbf{I}]\,\mathbf{P}_{\mathcal{G}}\,[\eta_1\mathcal{B}_1 + (1-\eta_1)\mathbf{I}]$, where $\mathbf{I}$ denotes the identity matrix. The effects of η_1 and η_2 which are set to a same value $\bar{\eta}$ is shown in Figure 5 . As the weight decreases, the accuracy increases first owing to better plasticity, and then decreases due to worse stability caused by the forgetting. It implies that a trade-off can be achieved by the two weights of projections.

Long-sequence Continual Learning We experiment on 5 benchmarks under the protocols of 50 tasks and 100 tasks to validate that our approach remains effective even within the context of long-sequence continual learning. The results are presented in Table 4. Despite lacking plasticity enhancement, VPT-NSP^2 can outperform existing state-of-the-art approaches and especially surpasses L2P by a large margin. This demonstrates that forgetting is still the predominant factor affecting performance in long sequence of tasks. With the plasticity enhancement, VPT-NSP^2 achieves significant increase in accuracy (by 1.1%∼2.9%). This demonstrates that our plasticity enhancement is effective in learning new knowledge in long-sequence continual learning.

https://doi.org/10.52202/079017-0253

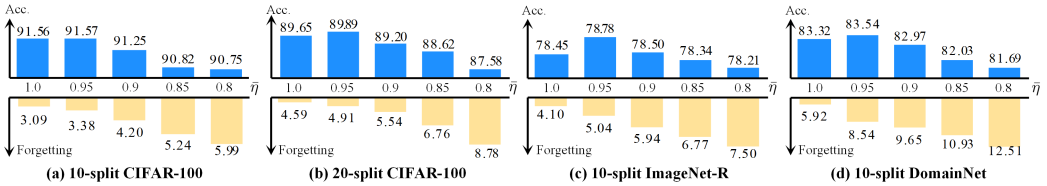

Figure 5: Effect of the projection matrix weight $\bar{\eta}$ on the accuracy and forgetting for the stability-plasticity trade-off on the four benchmarks.

Table 4: Results for 50 tasks and 100 tasks on CIFAR-100, ImageNet-R and DomainNet datasets. † indicates no plasticity enhancement, and ‡ indicates using the balanced plasticity enhancement where $\bar{\eta}$ is the default value less than 1. Our approach still outperforms other methods in long sequences of tasks.

Method	50S-CIFAR100		50S-ImageNet-R		50S-DomainNet		100S-ImageNet-R		100S-DomainNet	
	Acc.	Forgetting	Acc.	Forgetting	Acc.	Forgetting	Acc.	Forgetting	Acc.	Forgetting
L2P	76.19	12.06	48.53	12.99	59.45	11.53	38.87	15.26	50.52	17.66
EvoPrompt	<u>76.60</u>	13.86	<u>68.53</u>	10.03	67.68	10.41	<u>61.84</u>	15.84	<u>56.35</u>	21.39
OVOR	65.69	14.28	60.08	5.86	66.27	7.43	40.49	8.12	47.65	8.91
InfLoRA	61.49	13.68	59.02	11.02	<u>69.96</u>	9.51	38.16	15.11	44.32	17.85
EASE	74.47	9.31	68.17	7.76	61.20	10.01	47.55	8.22	33.08	32.14
CPrompt	74.97	7.45	68.47	8.16	67.87	9.36	56.95	10.20	53.73	12.14
VPT-Seq	70.47	29.21	56.38	37.91	58.39	44.79	49.72	45.53	46.39	49.34
VPT-NSP^{2†}	81.92	6.56	67.32	6.35	70.13	9.92	59.97	10.07	54.44	11.04
VPT-NSP^{2‡}	**82.98**	6.66	**69.48**	6.51	**71.28**	11.36	**62.23**	12.13	**57.35**	13.82

6 Conclusion

In this paper, we study the interference problem of visual prompt tuning in ViTs, and propose two consistency conditions which can eliminate the interference in theory under the constraint of invariant prompt distribution. They guarantee the consistency of Affinity, Aggregation and distribution of prompts in LayerNorm, respectively, which jointly achieve the consistency objective of the whole ViT layer. We adopt the null-space projection to implement the two conditions and utilize an extra loss to satisfy the constraint. Our experiments on various benchmarks demonstrate the effectiveness of the proposed conditions for anti-forgetting, and our approach achieves state-of-the-art performances.

Limitation Discussion: To simplify the derivation of our consistency conditions, we introduce a constraint of invariant prompt distribution. Although the superior results show that it may not be a very strong assumption, it is not an exact solution.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62101453, 62176198 and 62201467; in part by the Project funded by China Postdoctoral Science Foundation under Grant 2022TQ0260 and Grant 2023M742842, in part by the Young Talent Fund of Xi'an Association for Science and Technology under Grant 959202313088, in part by Innovation Capability Support Program of Shaanxi (Program No. 2024ZC-KJXX-043) and in part by the Natural Science Basic Research Program of Shaanxi Province (No. 2022JC-DW-08).

References

[1] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *CoRR*, abs/1607.06450, 2016.

[2] Benjamin Bowman, Alessandro Achille, Luca Zancato, Matthew Trager, Pramuditha Perera, Giovanni Paolini, and Stefano Soatto. À-la-carte Prompt Tuning (APT): Combining Distinct Data Via Composable Prompting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14984–14993, 2023.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 9630–9640, 2021.

[4] Arslan Chaudhry, Naeemullah Khan, Puneet K. Dokania, and Philip H. S. Torr. Continual Learning in Low-rank Orthogonal Subspaces. In *Advances in Neural Information Processing Systems*, 2020.

[5] Kumar Chellapilla, Sidd Puri, and Patrice Simard. High performance convolutional neural networks for document processing. In *Tenth international workshop on frontiers in handwriting recognition*. Suvisoft, 2006.

[6] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible Scaling Laws for Contrastive Language-Image Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[7] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Strategies From Data. In *IEEE/CVF International Conference on Computer Vision*, pages 113–123, 2019.

[8] Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening Sharpness for Dynamic Gradient Projection Memory Benefits Continual Learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 18710–18721, 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021.

[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 *IEEE/CVF International Conference on Computer Vision*, pages 11449–11459, 2023.

[11] Zhanxin Gao, Jun Cen, and Xiaobin Chang. Consistent Prompting for Rehearsal-Free Continual Learning. *CoRR*, abs/2403.08568, 2024.

[12] Wei-Cheng Huang, Chun-Fu Chen, and Hsiang Hsu. OVOR: OnePrompt with virtual outlier regularization for rehearsal-free class-incremental learning. In *International Conference on Learning Representations*, 2024.

[13] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning. In *European Conference on Computer Vision*, volume 13693, pages 709–727, 2022.

[14] Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating instance-level prompts for rehearsal-free continual learning. In *IEEE/CVF International Conference on Computer Vision*, pages 11813–11823, October 2023.

[15] Muhammad Gul Zain Ali Khan, Muhammad Ferjad Naeem, Luc Van Gool, Didier Stricker, Federico Tombari, and Muhammad Zeshan Afzal. Introducing language guidance in prompt-based continual learning. In *IEEE/CVF International Conference on Computer Vision*, pages 11429–11439, October 2023.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

[17] Yajing Kong, Liu Liu, Zhen Wang, and Dacheng Tao. Balancing Stability and Plasticity Through Advanced Null Space in Continual Learning. In *European Conference on Computer Vision*, volume 13686, pages 219–236, 2022.

 https://doi.org/10.52202/079017-0253

[18] Muhammad Rifki Kurniawan, Xiang Song, Zhiheng Ma, Yuhang He, Yihong Gong, Yang Qi, and Xing Wei. Evolving Parameterized Prompt Memory for Continual Learning. In *AAAI Conference on Artificial Intelligence*, pages 13301–13309, 2024.

[19] Zhuowei Li, Long Zhao, Zizhao Zhang, Han Zhang, Di Liu, Ting Liu, and Dimitris N. Metaxas. Steering Prototypes with Prompt-Tuning for Rehearsal-Free Continual Learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2523–2533, 2024.

[20] Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. *arXiv preprint arXiv:2404.00228*, 2024.

[21] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Elsevier, 1989.

[22] Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random Projections and Pre-trained Models for Continual Learning. In *Advances in Neural Information Processing Systems*, 2023.

[23] Thomas De Min, Massimiliano Mancini, Karteek Alahari, Xavier Alameda-Pineda, and Elisa Ricci. On the Effectiveness of LayerNorm Tuning for Continual Learning in Vision Transformers. In *IEEE/CVF International Conference on Computer Vision Workshops*, pages 3577–3586, 2023.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[25] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment Matching for Multi-Source Domain Adaptation. In *IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

[26] Jingyang Qiao, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, and Yuan Xie. Prompt Gradient Projection for Continual Learning. In *International Conference on Learning Representations*, 2024.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*, volume 139, pages 8748–8763, 2021.

[28] Roger Ratcliff. Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

[29] Tal Ridnik, Emanuel Ben Baruch, Asaf Noy, and Lihi Zelnik. ImageNet-21K Pretraining for the Masses. In *NeurIPS Datasets and Benchmarks*, 2021.

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[31] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient Projection Memory for Continual Learning. In *International Conference on Learning Representations*, 2021.

[32] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, June 2023.

[33] Yu-Ming Tang, Yi-Xing Peng, and Wei-Shi Zheng. When Prompt-based Incremental Learning Does Not Meet Strong Pretraining. In *IEEE/CVF International Conference on Computer Vision*, pages 1706–1716, 2023.

[34] Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical Decomposition of Prompt-Based Continual Learning: Rethinking Obscured Sub-optimality. In *Advances in Neural Information Processing Systems*, 2023.

[35] Runqi Wang, Xiaoyue Duan, Guoliang Kang, Jianzhuang Liu, Shaohui Lin, Songcen Xu, Jinhu Lv, and Baochang Zhang. AttriCLIP: A Non-Incremental Learner for Incremental Knowledge Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3654–3663, June 2023.

[36] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training Networks in Null Space of Feature Covariance for Continual Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 184–193, June 2021.

[37] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-Prompts Learning with Pre-trained Transformers: An Occam's Razor for Domain Incremental Learning. In *Advances in Neural Information Processing Systems*, 2022.

[38] 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 *AAAI Conference on Artificial Intelligence*, pages 10209–10217, 2023.

[39] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. DualPrompt: Complementary Prompting for Rehearsal-Free Continual Learning. In *European Conference on Computer Vision*, volume 13686, pages 631–648, 2022.

[40] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, June 2022.

[41] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[42] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, Peng Wang, and Yanning Zhang. Dual modality prompt tuning for vision-language pre-trained model. *IEEE Transactions on Multimedia*, 2023.

[43] Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual Learning of Context-Dependent Processing in Neural Networks. *Nature Machine Intelligence*, 1(8):364–372, August 2019.

[44] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow Learner with Classifier Alignment for Continual Learning on a Pre-trained Model. In *IEEE/CVF International Conference on Computer Vision*, pages 19091–19101, October 2023.

[45] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3718–3727, June 2023.

[46] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable Subspace Ensemble for Pre-Trained Model-Based Class-Incremental Learning. *CoRR*, abs/2403.12030, 2024.

 https://doi.org/10.52202/079017-0253

Appendix: Visual Prompt Tuning in Null Space for Continual Learning

A Algorithm

An overview and algorithm of our approach are provided in Figure 6 and Algorithm 1 , respectively. We first initialize the overall uncentered covariance matrices [36] $\mathbf{C}_1$ and $\mathbf{C}_2$, as well as the null-space projection matrices $\mathcal{B}_1$ and $\mathcal{B}_2$. During training, the cross-entropy loss for classification and the loss of prompt distribution $\mathcal{L}_{\mathrm{LN}}$ are jointly utilized for optimization. Subsequently, we get the candidate prompt updates $\mathbf{P}_{\mathcal{G}}$ computed by the optimizer. Then $\mathbf{P}_{\mathcal{G}}$ is projected by the null-space projection matrices $\mathcal{B}_1$ and $\mathcal{B}_2$ for updating the prompts. After the convergence, we obtain the matrices $\mathbf{J}_1$ and $\mathbf{J}_2$ to temporarily store $\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^{\top}$ and $\mathbf{S}_{\mathbf{P}_t}$ for the data of the current task. Then they are used to update the uncentered covariance matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ by addition. Finally, we update the null-space projection matrices using the uncentered covariance matrices, which will be used in the next task.

Algorithm 2 shows the process of computing a null-space projection matrix. First, an input uncentered covariance matrix $\mathbf{C}$ is decomposed by SVD, from which we can get the singular values and right singular vectors. Next, we determine the nullity R (*i.e.*, the dimension of null space) of $\mathbf{C}$ according to the maximum second derivative, which is introduced in Section C . Then we select R right singular vectors corresponding to the R smallest singular values considered close to 0 as the bases of null space. Finally, we compute the normalized projection matrix, which provides an upper bound for the scale of the projected gradients and prevents excessive gradient magnitudes. In our implementation, the null-space projection matrix is added by an identity matrix with a weight η (specifically η_1 for $\mathcal{B}_1$ and η_2 for $\mathcal{B}_2$). η is a hyper-parameter for the trade-off between stability and plasticity, which is also introduced in Section C

B Experimental Setups and Implementation Details

Models: We validate our approach on the Vision Transformer (ViT) [9] and CLIP [27] models in the experiments, whose backbones are both ViT-Base/16 [9]. The ViT is pre-trained on ImageNet-21k, and we insert 4 prompts into each of the 12 layers for fine-tuning, which is referred to as "VPT" [13]. The classifiers are dependently trained in each task and the orthogonal projection is not applicable to them. All the classifiers from the available tasks are concatenated to make prediction during inference. For the CLIP model pre-trained on the WebImageText, we insert 4 prompts into each of the first 3 layers of the image encoder, while the text encoder is kept frozen. The logit scale that serves as a learnable scalar parameter to scale the cosine similarities between image features and text features is also set to trainable. We observed a serious cross-task confusion among the tasks in the CLIP model. Hence, we follow [44] to utilize the class-wise mean and covariance of previous features extracted before the embedding projection head (*i.e.*, the last linear layer of the image encoder) to refine the projection head, after the prompt tuning stage in each task.

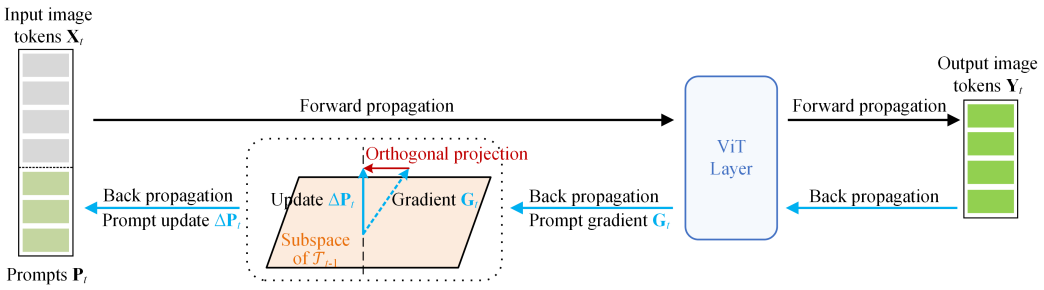

Figure 6: Illustration of our algorithm. The input image tokens with prompts are fed into the ViT layer for forward propagation. During optimization, the gradients of the prompts will be projected into the orthogonal direction to the subspace of the previous task $\mathcal{T}_{t-1}$. The projected prompt update will be used to update the prompts for anti-forgetting.

https://doi.org/10.52202/079017-0253

Algorithm 1 NSP^2 for Visual Prompt Tuning

Inputs: Datasets $\mathcal{D}_t = \{(\mathcal{X}_t^{<i>}, y_t^{<i>})\}_{i=1}^{|\mathcal{T}_t|}$ for task $\mathcal{T}_t \in \{\mathcal{T}_1, \mathcal{T}_2, \cdots\}$, ViT model $f_{\text{model}}(\cdot|\mathbf{P}_t)$ with the prompts $\mathbf{P}_t$ to be optimized (the classifier is omitted for simplicity), uncentered covariance matrices $\mathbf{C}_1$ and $\mathbf{C}_2$, projection matrices $\mathcal{B}_1$ and $\mathcal{B}_2$

Outputs: The optimized prompts $\mathbf{P}_t$

Initialization: Randomly initialize $\mathbf{P}_t$; $\mathbf{C}_1 = \mathbf{0}$, $\mathbf{C}_2 = \mathbf{0}$, $\mathcal{B}_1 = \mathbf{I}$, $\mathcal{B}_2 = \mathbf{I}$
for task $\mathcal{T}_t \in \{\mathcal{T}_1, \mathcal{T}_2, \cdots\}$ **do**
  repeat
    Sample a mini-batch $\boldsymbol{\mathcal{X}}_t, \boldsymbol{y}_t \sim \mathcal{D}_t$
    Obtain prediction by $\hat{\boldsymbol{y}}_t \leftarrow f_{\text{model}}(\boldsymbol{\mathcal{X}}_t|\mathbf{P}_t)$
    Compute the classification loss $\mathcal{L}_{total} \leftarrow \text{CrossEntropy}(\hat{\boldsymbol{y}}_t, \boldsymbol{y}_t)$
    if $t > 1$ **then**
      Compute the loss of prompt distribution $\mathcal{L}_{\text{LN}}$ by Eq. (27)
      Accumulate the losses $\mathcal{L}_{total} \leftarrow \mathcal{L}_{total} + \mathcal{L}_{\text{LN}}$
    end if
    Get the candidate prompt update $\mathbf{P}_{\mathcal{G}}$ from the optimizer by the loss $\mathcal{L}_{total}$
    if $t > 1$ **then**
      Compute the prompt update $\Delta\mathbf{P} \leftarrow \mathcal{B}_2\mathbf{P}_{\mathcal{G}}\mathcal{B}_1$ by the null-space projection Eq. (26)
    else
      Directly adopt the candidate prompt update $\Delta\mathbf{P} \leftarrow \mathbf{P}_{\mathcal{G}}$
    end if
    Update the prompts by $\mathbf{P}_t \leftarrow \mathbf{P}_t - learning\_rate \times \Delta\mathbf{P}$
  until convergence
  Initialize two temporary matrices $\mathbf{J}_1 = [\,]$ and $\mathbf{J}_2 = [\,]$
  for $\mathcal{X}_t^{<i>} \in \mathcal{D}_t$ **do**
    Get the matrices $(\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top)^{<i>}$ and $\mathbf{S}_{\mathbf{P}_t}^{<i>}$ by the forward propagation $f_{\text{model}}(\mathcal{X}_t^{<i>}|\mathbf{P}_t)$
    Update $\mathbf{J}_1$ and $\mathbf{J}_2$ by concatenating $(\mathbf{Q}_{\mathbf{X}_t}\mathbf{W}_k^\top)^{<i>}$ and $\mathbf{J}_1$, $\mathbf{S}_{\mathbf{P}_t}^{<i>}$ and $\mathbf{J}_2$, respectively
  end for
  Update the uncentered covariance matrices $\mathbf{C}_1 \leftarrow \mathbf{C}_1 + \mathbf{J}_1^\top\mathbf{J}_1$ and $\mathbf{C}_2 \leftarrow \mathbf{C}_2 + \mathbf{J}_2^\top\mathbf{J}_2$
  Compute the null-space projection matrices $\mathcal{B}_1$ and $\mathcal{B}_2$ by Algorithm 2 using $\mathbf{C}_1$ and $\mathbf{C}_2$
end for

Algorithm 2 Computing Null-Space Projection Matrix

Inputs: Uncentered covariance matrix $\mathbf{C}$, hyper-parameter $\eta \in [0, 1]$ for the trade-off between stability and plasticity (mentioned in Section C)

Outputs: Null-space projection matrix $\mathcal{B}$

Get the singular values Λ in descending order and the corresponding right singular vectors $\mathbf{U}$ by singular value decomposition $\Lambda, \mathbf{U}^\top \leftarrow \text{SVD}(\mathbf{C})$, where the left singular vectors are omitted
Calculate the nullity R by the maximum second derivative as introduced in Eq. (28)
Select the right singular vectors of the R smallest singular values in $\mathbf{U}$ as $\mathbf{U}_0 \leftarrow \mathbf{U}_{[D-R:D]}$
Compute the projection matrix $\mathcal{B} \leftarrow \frac{\mathbf{U}_0\mathbf{U}_0^\top}{\|\mathbf{U}_0\mathbf{U}_0^\top\|_{\text{F}}}$
Update $\mathcal{B}$ with the weight η by $\mathcal{B} \leftarrow \eta\mathcal{B} + (1-\eta)\mathbf{I}$ (corresponding to Eq. (29))

Benchmarks: We conduct experiments under the class-incremental learning protocol, where the classes in each task are disjoint, and task identity is unknown during inference. Four class-incremental benchmarks with three widely used datasets are adopted: 10- and 20-split CIFAR-100, 10-split ImageNet-R [39] and 10-split DomainNet [25, 38]. For the CIFAR-100 dataset, the total of 100 classes are randomly split into 10 or 20 tasks, which can evaluate the ability to handle different numbers of tasks. We follow [39] to randomly split the 200 classes in ImageNet-R into 10 tasks, which forms the 10-split ImageNet-R benchmark. For the 10-split DomainNet, we follow the same dataset protocol adopted in [38] and [11] to select the top 200 classes with the most images from the original DomainNet [25], and randomly split them into 10 tasks with 20 classes per task. 25% samples of the training data in each dataset are picked as a validation set for searching optimal hyper-parameters.

https://doi.org/10.52202/079017-0253

Metrics: Formally, the final average accuracy and final average forgetting are defined as:

$$\text{Final average accuracy} = \frac{1}{T}\sum_{i=1}^{T} a_{T,i},$$

$$\text{Final average forgetting} = \frac{1}{T-1}\sum_{i=1}^{T-1} \max_{j\in\{1,2,\cdots,T-1\}} (a_{j,i} - a_{T,i}),$$

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

Higher accuracy means the model performs better, while lower forgetting means stronger stability (*i.e.*, the ability to retain old knowledge). However, lower forgetting does not always generate higher accuracy since the accuracy is also affected by plasticity (*i.e.*, the ability to learn new knowledge). The accuracy is the main metric we should focus on as it reflects the precision of classification in practice.

Implementations Details: For all the datasets and models, the images fed into the models are resized to 224×224 pixels and augmented by AutoAugment [7] during training. For the VPT-based models, we use the Adam optimizer [16] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a weight decay of 5×10^{-5} to train 100 epochs with an initial learning rate of 0.01 and a batch size of 256 on all benchmarks. The learning rate is scaled by a factor of 0.1 at the 50-th and 80-th epoch. Our training losses consist of the cross-entropy loss for classification and the loss $\mathcal{L}_{\text{LN}}$ in Eq. (27) whose coefficient is set to 1. Through cross validation on the validation set, we set the temperatures in the cross-entropy loss to 28, 25, 30 and 30 for the 10-split CIFAR100, 20-split CIFAR100, 10-split ImageNet-R and 10-split DomainNet benchmarks. There are two hyper-parameters η_1 and η_2 used for the trade-off between stability and plasticity in null-space projection as introduced in Section C , and we set both of them to be 0.97, 0.95, 0.94 and 0.95 for the four benchmarks by cross validation.

As to the CLIP-based models, the differences in training settings are as follows. We train them for 20 epochs with the batch size of 220 and the learning rate 0.001 which decays at the 10-th and 16-th epoch. The temperatures are all set to 1 since the logit scale is trainable. η_1 and η_2 are set to 0.98 which is a proper value for all the benchmarks. We refine the embedding projection head for 50 epochs using the SGD optimizer with a learning rate of 0.001, a momentum of 0.9 and a weight decay of 1×10^{-4}.

We implement our approach in PyTorch [24] with the timm library [41]. The experiments are performed on a server with 128 GB RAM and four NVIDIA RTX 4090 GPUs. Each of the experiment can be finished in three hours.

C Trade-off between Stability and Plasticity

Given that the null space of covariance matrix does not always exist in practice, Wang *et al.* [36] suggest approximating it by selecting the bases whose associated singular values approach zero, where the singular values smaller than a specified multiple (denoted as γ in our paper) of the smallest one are selected. However, we experimentally find this strategy and the experience for selecting γ are not suitable for prompt tuning in ViTs to determine the nullities R_1 and R_2 for the uncentered covariance matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ in Algorithm 1 , which will be introduced afterwards. To solve this problem, we propose an adaptive nullity strategy to determine the nullities in an adaptive manner. Utilizing the characteristic that the curve of descending singular values forms an "L" shape, we divide the curve into two parts by the point where the gradient changes fastest to cover most of the small singular values. It is realized by calculating the maximum second derivative of the points:

$$\begin{cases} R_1 = D - \arg\max_j \{\lambda_{j-1} - 2\lambda_j + \lambda_{j+1}\}_{j=2}^{D-1}, \\ R_2 = M - \arg\max_j \{\lambda_{j-1} - 2\lambda_j + \lambda_{j+1}\}_{j=2}^{M-1}, \end{cases} \tag{28}$$

where λ_j denotes the j-th singular value. We find it reaches near-minimum forgetting in our experiments which also means reaching near-optimal stability. Furthermore, to enhance the plasticity, we fuse the projection matrices with identity matrices by the weights $\eta_1 \in [0, 1]$ and $\eta_2 \in [0, 1]$ which should be close to 1:

$$\Delta\mathbf{P} = [\eta_2\mathcal{B}_2 + (1-\eta_2)\mathbf{I}]\,\mathbf{P}_{\mathcal{G}}\,[\eta_1\mathcal{B}_1 + (1-\eta_1)\mathbf{I}]\,. \tag{29}$$

In this way, we can make a trade-off between stability and plasticity by enhancing the plasticity based on near-optimal stability, and η_1 and η_2 are the hyper-parameters to control the trade-off.

D Comparison with PGP

D.1 Difference in Methods

The main difference between our method and PGP [26] are summarized as follows. (1) We derive a different consistency condition for Affinity even if we ignore the LayerNorm operation and the bias terms in the qkv-transformation. Specifically, our simplified consistency condition for Affinity is $\mathbf{X}_t\mathbf{W}_q\mathbf{W}_k^\top\Delta\mathbf{P}^\top = \mathbf{0}$, contrasted with $\mathbf{X}_t\Delta\mathbf{P}^\top = \mathbf{0}$ in PGP. (2) We analyze the consistency conditions for the complete self-attention, *i.e.*, $\mathrm{softmax}(\frac{\mathbf{Q_X}\mathbf{K_Z}^\top}{\sqrt{D}})\mathbf{V_Z}$ which contains the Aggregation operation. However, PGP does not account for the Aggregation. (3) We take the LayerNorm before self-attention into consideration and propose an invariant prompt distribution constraint, while it is ignored in PGP.

In conclusion, we conduct a comprehensive analysis of prompt tuning for the consistency objective, which provides a complete guarantee to eliminate the interference of new tasks on previous tasks. As demonstrated in our ablation study in the Experiment section, the consistency of Aggregation and LayerNorm also contribute to reducing forgetting, and thereby they should not be ignored. We make a comparison of the performance between PGP and our approach in the next subsection.

D.2 Performance Comparison

We compare with PGP [26] using the VPT-Seq and L2P [40] baselines on the four benchmarks in our experiments. The results are shown in Table 5 . We implement PGP to VPT (*i.e.* VPT-PGP) under the same training settings as VPT-NSP^2 for a fair comparison. For the L2P-based methods, we insert prompts into the first three layers instead of only the first layer in the original implementation [40]. An orthogonal projection is also applied to the prompt pool which is essentially a linear layer in L2P-based models. We follow the training setting of PGP to train the L2P-based methods. The results in Table 5 demonstrate that our full approach can achieve more improvements in accuracy and reduce more forgetting than PGP. Even when applying only the projection matrix $\mathcal{B}_1$ for the Affinity operation, our approach also performs better than PGP, demonstrating the effectiveness of our proposed method for mitigating the interference problem.

Table 5: Comparison with PGP on four benchmarks and two continual learning baselines. "-$\mathcal{B}_1$" indicates only the projection matrix $\mathcal{B}_1$ is used in our approach

Method	10S-CIFAR-100		20S-CIFAR-100		10S-ImageNet-R		10S-DomainNet	
	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓	Acc.↑	Forgetting↓
VPT-Seq	87.27	12.33	82.36	17.36	72.46	19.41	73.28	25.65
VPT-PGP	87.76	11.98	82.71	16.85	73.12	18.92	73.98	25.15
VPT-NSP^2-$\mathcal{B}_1$	90.58	6.91	88.13	10.27	78.05	8.14	82.31	10.89
VPT-NSP^2	**91.74**	3.28	**89.89**	4.91	**78.88**	5.06	**83.54**	8.54
L2P	84.12	6.36	81.46	8.69	61.25	9.32	65.73	10.19
L2P-PGP	84.70	5.96	82.04	8.11	62.01	8.55	66.31	9.63
L2P-NSP^2-$\mathcal{B}_1$	86.39	4.60	82.99	7.34	64.10	7.17	67.48	8.21
L2P-NSP^2	**86.78**	4.22	**83.37**	6.93	**64.66**	6.84	**68.14**	7.79

https://doi.org/10.52202/079017-0253

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state the claims in our abstract: we derive two consistency conditions of eliminating the interference problem under the invariant prompt distribution assumption for visual prompt tuning in the filed of continual learning. We implement them by the null-space projection method, and we validate the effectiveness and generalizability of our method. Our contributions are elaborated in the last paragraph of the Introduction section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitation of our approach is discussed in the Conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The formulas used to derive our proposed conditions are numbered or cross-referenced.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the algorithm of our approach, experimental settings and hyper-parameters adopted in our experiments in the Experimental Setups and Implementation Details section of the appendix. Our code is also available in the supplemental material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

https://doi.org/10.52202/079017-0253

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code is available in the supplemental material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the detailed experimental settings, including the data splits, hyperparameters, optimizer and other settings in the experimental setups of appendix. We also provide the code in the supplemental material for a thorough reference.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the mean results over three runs in our experiments, and we report the standard deviations in the subsection of comparison with existing methods.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the information of our compute resources in the experimental setups of appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and make sure our research conforms the ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

https://doi.org/10.52202/079017-0253

Justification: As a fundamental research in machine learning, the potential societal impact is not obvious at this stage.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No new pre-trained models or datasets are released in this work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We explicitly cite the used assets in our papers, including the ViT model, timm library, DomainNet dataset, *etc.*, and respect their license and terms during usage.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code is provided in the supplemental material. A documentation for running the experiments is contained in the code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

https://doi.org/10.52202/079017-0253

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.