Rethinking Gradient Projection Continual Learning: Stability / Plasticity Feature Space Decoupling

Zhen Zhao[1], Zhizhong Zhang[1,†], Xin Tan[1], Jun Liu[2], Yanyun Qu[3], Yuan Xie[1,†], Lizhuang Ma[1]
[1]School of Computer Science and Technology
East China Normal University, Shanghai, China
[2]Tencent Youtu Lab [3]School of Informatics, Xiamen University, Fujian, China
{51255901056}@stu.ecnu.edu.cn, {zzzhang,xtan,yxie,lzma}@cs.ecnu.edu.cn
{junsenselee}@tencent.com, {yyqu}@xmu.edu.cn

Abstract

Continual learning aims to incrementally learn novel classes over time, while not forgetting the learned knowledge. Recent studies have found that learning would not forget if the updated gradient is orthogonal to the feature space. However, previous approaches require the gradient to be fully orthogonal to the whole feature space, leading to poor plasticity, as the feasible gradient direction becomes narrow when the tasks continually come, i.e., feature space is unlimitedly expanded. In this paper, we propose a space decoupling (SD) algorithm to decouple the feature space into a pair of complementary subspaces, i.e., the stability space $\mathcal{I}$, and the plasticity space $\mathcal{R}$. $\mathcal{I}$ is established by conducting space intersection between the historic and current feature space, and thus $\mathcal{I}$ contains more task-shared bases. $\mathcal{R}$ is constructed by seeking the orthogonal complementary subspace of $\mathcal{I}$, and thus $\mathcal{R}$ mainly contains task-specific bases. By putting distinguishing constraints on $\mathcal{R}$ and $\mathcal{I}$, our method achieves a better balance between stability and plasticity. Extensive experiments are conducted by applying SD to gradient projection baselines, and show SD is model-agnostic and achieves SOTA results on publicly available datasets.

1. Introduction

Deep neural networks (DNNs) have achieved promising performance on various vision tasks, including image classification, object detection, and action recognition [3, 9, 32, 34, 36]. However, DNNs are typically trained offline on a fixed dataset, and therefore the models are not able to incrementally learn novel concepts (novel classes), which has become an emerging need in many real-world applications [11, 16, 21, 26, 30].

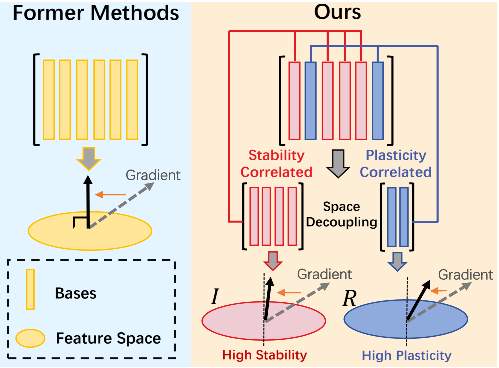

Figure 1. Left: Recent gradient projection methods. All of them constrain the gradient to be fully orthogonal to the feature space. Right: We propose a space decoupling (SD) algorithm to decouple the feature space into a pair of complementary subspaces, *i.e.*, the stability space $\mathcal{I}$, and the plasticity space $\mathcal{R}$. To balance stability and plasticity, more bases are preserved in $\mathcal{I}$, and less in $\mathcal{R}$, while stricter gradient constraints are put on $\mathcal{I}$ and looser on $\mathcal{R}$.

In this context, continual learning (CL) [14] is proposed, aiming to continually learn novel concepts, *i.e.*, a series of learning tasks, while not forgetting the learned knowledge [1, 4, 6, 10, 15, 39]. Recent studies have found that learning would have less impact on old tasks if the direction of the gradient is orthogonal to the space spanned by the features from old tasks [18, 23, 31, 38]. With this motivation, a couple of continual learning methods referring to feature space methods have been proposed and can be generally divided into two classes: (a) **Orthogonal based** methods; (b) **Null-space based** methods.

Orthogonal based methods like GPM [31] and TRGP [23] calibrate the gradient in the direction fully orthogonal to the feature space, while Null-space based methods like Adam-NSCL [38], AdNS [18] train the model in the null space of input features. It is easy to prove that these

[†] Corresponding authors.

two classes of approaches are equivalent and hold a unified training paradigm: 1) construct a matrix using the features from old tasks, *e.g.,* concatenate; 2) utilize this matrix to approximate a feature space; 3) project the gradient of the new task to the orthogonal direction of the feature space.

However, we find all the mentioned approaches strictly require the gradient to be fully orthogonal to the whole feature space, shown in the left of Figure 1. As the number of training tasks increases, feature space is unlimitedly expanded which will heavily limit the model updating and lead to poor plasticity. Therefore, feature space methods are facing a dilemma in balancing stability and plasticity [27–29, 33, 40], despite their varied attempts in this issue.

Motivated by this insight, we propose a space decoupling (SD) algorithm, shown in the right of Figure 1. We decouple the whole feature space into a pair of orthogonal complementary subspaces, *i.e.*, the stability-correlated space $\mathcal{I}$, and the plasticity-correlated space $\mathcal{R}$. In our implementations, $\mathcal{I}$ is established by conducting space intersection between the historic feature space and current feature space, and thus $\mathcal{I}$ contains more bases shared by old tasks. $\mathcal{R}$ is constructed by seeking the orthogonal complementary subspace of $\mathcal{I}$, and thus $\mathcal{R}$ mainly contains task-specific bases. As we can see, the update on $\mathcal{I}$ would significantly incur forgetting, and the update on $\mathcal{R}$ would have less impact on old tasks. Our empirical study also supports this claim by finding that gradient updates within subspace $\mathcal{I}$ do more interference on old tasks than $\mathcal{R}$ (please refer to Section 3.2).

Finally, in the stability-correlated space $\mathcal{I}$, where a slight change would bring about tremendous forgetting, we pay more attention to stability by putting more strict constraints on it. In the plasticity-correlated space $\mathcal{R}$ which will have less impact on old knowledge, we stress plasticity and allow the model to be updated in a looser way here. Finally, with SD, the performance of several state-of-the-art gradient projection methods is improved by a large margin. Below, we summarize our contributions:

(1) We generalize recent gradient projection methods [18, 23, 31, 38] into a unified paradigm, under which we give a new viewpoint about their stability-plasticity dilemma.

(2) We propose a novel Space Decoupling (SD) algorithm to split the whole feature space into stability-correlated space and plasticity-correlated space. By putting distinguishing constraints on these subspaces, our method achieves a better balance between stability and plasticity.

(3) We apply SD to various gradient projection baselines and show our approach is model-agnostic and effective. Extensive experiments on benchmark datasets demonstrate state-of-the-art performance achieved by our approach.

2. Related Work and Preliminaries

To reduce the interference to old tasks, recent studies [18, 23, 31, 38] have focused on leveraging the feature space to modify the gradient, such that the parameter update is along the direction with less impact on old task. Representative works including **Orthogonal based** methods which constrain the gradient to be orthogonal to the feature space, *i.e.*, GPM [31] and TRGP [23], and **Null-space based** methods which update the model in the null space of the features' uncentered covariance, *i.e.*, Adam-NSCL [38] and AdNS [18]. It can be easily proved that updating the model in the null space is equivalent to updating the model in the direction orthogonal to the feature space (please refer to supplementary materials), thus all of our subsequent discussions are based on the orthogonal algorithm. In what follows, we will briefly introduce the notations and preliminaries used in this paper.

Algorithm 1: Subspace Intersection

```
Input: $\mathcal{P} = span\{\boldsymbol{P}\}$, $\mathcal{Q} = span\{\boldsymbol{Q}\}$, where
       $\boldsymbol{P} \in \mathbb{R}^{d\times k_1}$, $\boldsymbol{Q} \in \mathbb{R}^{d\times k_2}$
Output: Subspace intersection $\mathcal{I} = span\{\boldsymbol{I}\}$, where $\boldsymbol{I} \in \mathbb{R}^{d\times k}$
$\boldsymbol{A} \leftarrow [\boldsymbol{P}, -\boldsymbol{Q}] \in \mathbb{R}^{d\times(k_1+k_2)}$
Solve homogeneous linear equation $\boldsymbol{A}\cdot\boldsymbol{X} = \boldsymbol{0}$
  and get basic solutions $\boldsymbol{N} \in \mathbb{R}^{(k_1+k_2)\times k}$
$\boldsymbol{I} \leftarrow \boldsymbol{P}\cdot\boldsymbol{N}[0:k_1] \in \mathbb{R}^{d\times k}$
Return $\mathcal{I} = span\{\boldsymbol{I}\}$.
```

2.1. Notations

Continual Learning. In continual learning, a network f parameterized by $\mathbb{W} = \{\boldsymbol{\theta}^l\}_{l=1}^L$ is sequentially trained on a stream of tasks $\mathbb{T} = \{t\}_{t=1}^T$. Each task t has a dataset $\mathbb{D}_t = \{(\boldsymbol{x}_{t,i}, y_{t,i})\}_{i=1}^{n_t}$ of size n_t, where $\boldsymbol{x}_{t,i}$ denotes the input vector and $y_{t,i}$ denotes the label. The learnt model after training the t-th task is parameterized by $\mathbb{W}_t = \{\boldsymbol{\theta}_t^l\}_{l=1}^L$. The feature for layer l is represented as $\boldsymbol{x}_{t,i}^l$ and $\boldsymbol{x}_{t,i}^1 = \boldsymbol{x}_{t,i}$. Denote $\mathcal{L}_t = \mathcal{L}_t(\mathbb{W}, \mathbb{D}_t)$ (*e.g.,* cross-entropy loss) as the loss function for task t.

Feature Subspace. We use $\mathcal{S} = span\{\boldsymbol{B}\}$ to represent a subspace in a d-dimensional space, where $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_k] \in \mathbb{R}^{(d\times k)}$ is the bases for $\mathcal{S}$. In the context of continual learning, we use $\mathcal{S}_t^l$ to denote the feature subspace spanned by the inputs of task t for layer l. It is clear that $\boldsymbol{x}_{t,i}^l \in \mathcal{S}_t^l$. And, for any matrix $\boldsymbol{A}$ whose row vector is d-dimensional, the projection of $\boldsymbol{A}$ onto $\mathcal{S}$ is defined as:

$$\text{Proj}_{\mathcal{S}}(\boldsymbol{A}) = \boldsymbol{A}\boldsymbol{B}(\boldsymbol{B}^{\text{T}}). \tag{1}$$

2.2. Subspace Intersection and Sum

Consider two subspaces $\mathcal{P} = span\{\boldsymbol{P}\}$, $\mathcal{Q} = span\{\boldsymbol{Q}\}$ in a d-dimensional space, the intersection and sum between $\mathcal{P}$ and $\mathcal{Q}$ [35, pg.459-460] are mathematically defined as:

$$\begin{aligned} \mathcal{P}\cap\mathcal{Q} &= \{\boldsymbol{\alpha} | \boldsymbol{\alpha}\in\mathcal{P}, \boldsymbol{\alpha}\in\mathcal{Q}\} \\ \mathcal{P}+\mathcal{Q} &= \{\boldsymbol{\alpha}+\boldsymbol{\beta} | \boldsymbol{\alpha}\in\mathcal{P}, \boldsymbol{\beta}\in\mathcal{Q}\}. \end{aligned} \tag{2}$$

```
Algorithm 2: Subspace Sum
Input: $\mathcal{P} = span\{\boldsymbol{P}\}, \mathcal{Q} = span\{\boldsymbol{Q}\}$, where
       $\boldsymbol{P} \in \mathbb{R}^{d\times k_1}, \boldsymbol{Q} \in \mathbb{R}^{d\times k_2}$
Output: Subspace sum $\mathcal{S} = span\{\boldsymbol{S}\}$, where $\boldsymbol{S} \in \mathbb{R}^{d\times k}$
$\hat{\boldsymbol{Q}} \leftarrow \boldsymbol{Q} - (\boldsymbol{P}\boldsymbol{P}^T)\boldsymbol{Q}$
Orthogonalize $\hat{\boldsymbol{Q}}$ using SVD.
$\boldsymbol{S} \leftarrow [\boldsymbol{P}, \hat{\boldsymbol{Q}}]$
Return $\mathcal{S} = span\{\boldsymbol{S}\}$.
```

The solution for seeking the intersection of two subspaces can be reduced to solve a homogeneous linear equation, while to find the subspace sum, GPM [31] gives a simple yet effective projection solution. Here we directly give Algorithm 1 and Algorithm 2 to briefly summarize these processes. The proof is presented in supplementary materials.

3. Method

In this section, we first show that existing feature space methods can be summarized in a generalized training paradigm. By deeply analyzing, we find there are critical parameters influencing stability and plasticity, and therefore propose a new decoupling algorithm.

3.1. Feature Space Continual Learning Paradigm

Let us start from GPM [31], a representative feature space method. When learning the t-th task, GPM updates the model $\mathbb{W}_{t-1} = \{\boldsymbol{\theta}^l_{t-1}\}^L_{l=1}$ by projecting the gradient $\nabla_{\boldsymbol{\theta}^l}\mathcal{L}_t$ onto the orthogonal direction of $\{\mathcal{S}^l_j\}^{t-1}_{j=1}$ using Eq. (1). Let $x^l_{j,i}$ denote the features extracted from the j-th task (where $j < t$), and $\Delta\boldsymbol{\theta}^l_{t-1}$ denote the parameter changes after learning the t-th task. Then we have:

$$\begin{aligned} \boldsymbol{\theta}^l_t x^l_{j,i} &= (\boldsymbol{\theta}^l_{t-1} + \Delta\boldsymbol{\theta}^l_{t-1})x^l_{j,i} \\ &= \boldsymbol{\theta}^l_{t-1}x^l_{j,i} + \Delta\boldsymbol{\theta}^l_{t-1}x^l_{j,i} \\ &= \boldsymbol{\theta}^l_{t-1}x^l_{j,i}. \end{aligned} \quad (3)$$

The above equation implies that old tasks suffer from no interference after the gradient projection.

As is shown in Figure 2, by carefully analyzing the gradient constraints of feature space [18, 23, 31, 38], we find all of them can be generalized into a unified paradigm despite their own small modifications (Please refer to supplementary materials for detailed analysis and proof). Some critical common steps are listed as follows:

1) Feature Matrix Construction. At the end of task t, feature space methods obtain layer-wise feature matrix $\boldsymbol{M}^l_t$ by using the data from the current task with a construction strategy $\mathcal{P}$:

$$\boldsymbol{M}^l_t = \mathcal{P}(\mathbb{D}_t; \mathbb{W}_t) \quad (4)$$

where $\mathcal{P}$ refers to random selection for [23, 31] and uncentered covariance for [18, 38]. For example, GPM [31] and TRGP [23] randomly select n samples from $\mathbb{D}_t$ and concatenate the input features together to obtain the feature matrix $\boldsymbol{M}^l_t = [x^l_{t,1}, x^l_{t,2}, \cdots, x^l_{t,n}]$.

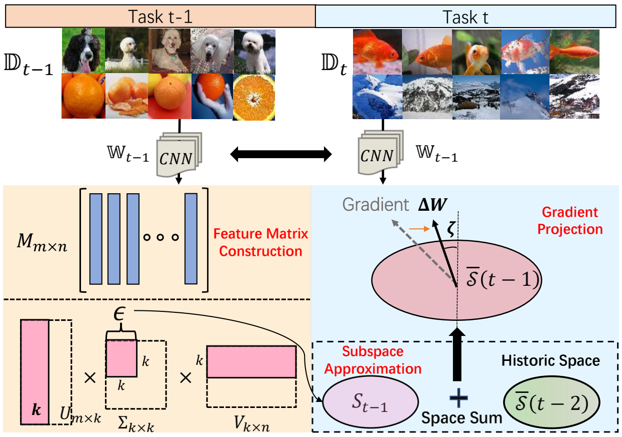

Figure 2. Feature Space Continual Learning Paradigm. (1) Construct feature matrix M after the training of the old task. (2) Approximate M and obtain the feature subspace $\mathcal{S}_{t-1}$. (3) Project new gradients to the orthogonal direction of the whole feature space $\bar{\mathcal{S}}(t-1)$. We demonstrate that the feature space dimension parameter ϵ and the gradient projection degree parameter ζ would influence stability and plasticity. Here the historic space $\bar{\mathcal{S}}(t)$ is the whole space of learned tasks, later defined in Eq.(6).

2) Subspace Approximation. Then, an approximation strategy $\mathcal{A}$ with hyper-parameter ϵ is adopted to obtain bases for the t-th task feature subspace $\boldsymbol{B}^l_t$:

$$\boldsymbol{B}^l_t = \mathcal{A}(\boldsymbol{M}^l_t; \epsilon). \quad (5)$$

The t-th feature subspace is then represented as $\mathcal{S}^l_t = span\{\boldsymbol{B}^l_t\}$. The strategy $\mathcal{A}$ usually represents Singular Value Decomposition (SVD). Suppose $\boldsymbol{M}^l_t \in \mathbb{R}^{m\times n}$, by performing SVD on $\boldsymbol{M}^l_t$, we have $\boldsymbol{M}^l_t = \boldsymbol{U}^l\boldsymbol{\Sigma}^l(\boldsymbol{V}^l)^T$, where $\boldsymbol{U}^l \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V}^l \in \mathbb{R}^{n\times n}$ are orthogonal, and $\boldsymbol{\Sigma}^l \in \mathbb{R}^{m\times m}$ contains the sorted singular values along its main diagonal [7]. GPM [31] and TRGP [23] pick k singular vectors with the top-k largest singular values in $\boldsymbol{U}^l$ to form $\boldsymbol{B}^l_t$ such that $\left\|(\boldsymbol{\Sigma}^l)_k\right\|^2_F \geq \epsilon \cdot \left\|\boldsymbol{\Sigma}^l\right\|^2_F$ holds, where $(\boldsymbol{\Sigma}^l)_k \in \mathbb{R}^{k\times k}$ contains the top-k largest singular values along its main diagonal and $\|\cdot\|_F$ is the Frobenius norm. As a result, the larger ϵ is, the higher the dimension of the feature space is obtained.

3) Gradient Projection. After subspace approximation, the whole feature space can be represented as:

$$\bar{\mathcal{S}}^l(t) = \mathcal{S}^l_1 + \mathcal{S}^l_2 + \cdots + \mathcal{S}^l_t. \quad (6)$$

Then the layer-wise gradient $\nabla_{\boldsymbol{w}^l}\mathcal{L}_{t+1}$ is constrained by a projection strategy $\mathcal{C}$ with hyper-parameter ζ to avoid forgetting:

$$\nabla_{\boldsymbol{w}^l}\mathcal{L}_{t+1} = \mathcal{C}(\nabla_{\boldsymbol{w}^l}\mathcal{L}_{t+1}; \zeta, \bar{\mathcal{S}}^l(t)). \quad (7)$$

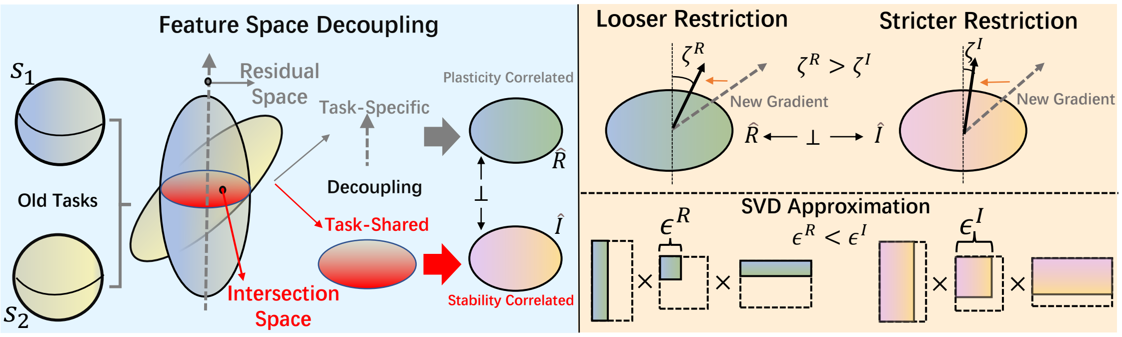

Figure 3. Left: Our proposed Space Decoupling (SD) algorithm. We decouple the whole feature space to a pair of complementary subspaces, *i.e.*, stability space $\mathcal{I}$ and plasticity space $\mathcal{R}$. $\mathcal{I}$ mainly contains task-shared bases, while $\mathcal{R}$ mainly contains task-specific bases. Right: We put distinguishing constraints on $\mathcal{I}$ and $\mathcal{R}$. For $\mathcal{I}$ more bases are preserved and the gradient restriction is stricter, which guarantees stability; For $\mathcal{R}$ fewer bases are preserved and the gradient restriction is looser, which guarantees plasticity.

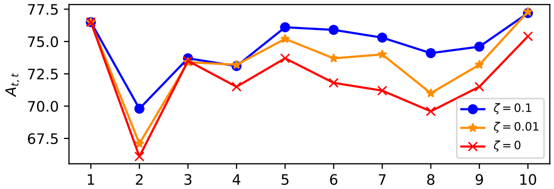

Figure 4. Accuracy of the t-th task on the t-th incremental session ($A_{t,t}$) of GPM [31] on 10-split-CIFAR100 with varied ζ. It is shown that a slight angle ζ implies higher plasticity.

Under the guidance of Eq.(3) [18,23,31,38], the gradient is restricted to be fully orthogonal to $\bar{\mathcal{S}}^l(t)$. Here we set ζ as a hyper-parameter to control the degree of orthogonality. For example, the gradient could be instantiated as:

$$\nabla_{\boldsymbol{w}^l}\mathcal{L}_{t+1} = \nabla_{\boldsymbol{w}^l}\mathcal{L}_{t+1} - \nabla_{\boldsymbol{w}^l}\mathcal{L}_{t+1}(1-\zeta)\bar{\mathcal{S}}^l(t)(\bar{\mathcal{S}}^l(t))^T \tag{8}$$

where ζ is zero, *i.e.,* a full orthogonality in previous works. In subsequent discussions, we remove the layer-wise notation for simplicity of expression.

Balancing between Stability and Plasticity. From this paradigm, we can easily find that there exist two factors that influence stability and plasticity. Firstly, ϵ at the approximation stage controls the dimension of the feature space. GPM [31] speculates that the higher dimension of approximated feature space is, the better the stability and the worse the plasticity will be. Secondly, ζ controls the degree of orthogonality. Although recent feature space methods constrain the updates in a fully orthogonal manner, we argue that a slight angle ζ would be a better choice as a full orthogonal manner is too strict and severely harms the model's plasticity, shown in Figure 4.

From the above analysis, we can observe one major deficiency that all of the gradient projection approaches treat the task-specific feature subspaces $\mathcal{S}_t$ in the same manner (uniform ϵ and ζ), but, as we can see, the stability and plasticity largely depend on the learned feature space . Therefore, in the next, we will give a new viewpoint about stability and plasticity from a feature space decoupling perspective.

3.2. Space Decoupling: Stability and Plasticity

As is shown in the left of Figure 3, we find two subspaces, namely intersection and residual subspaces in feature space $\bar{\mathcal{S}}(t)$ play a very important role for stability and plasticity. Formally, the intersection subspace is defined as the sum of all the intersection subspaces, *i.e.*,

$$\mathcal{I}(t) = \sum_{1<i\leq t} \bar{\mathcal{S}}(i-1) \cap \mathcal{S}_i. \tag{9}$$

Residual subspace is the orthogonal complement space of $\mathcal{I}(t)$, and therefore we have $\bar{\mathcal{S}}(t) = \mathcal{I}(t) + \mathcal{R}(t)$, and $\mathcal{I}(t)$ and $\mathcal{R}(t)$ are orthogonal.

We say $\mathcal{I}(t)$ is the intersection subspace as it contains the bases shared by multiple old tasks, while we say $\mathcal{R}(t)$ is the residual subspace as it mainly contains task-specific bases. From this perspective, we can see $\mathcal{I}(t)$ has more correlation to stability since the update on $\mathcal{I}(t)$ would significantly incur forgetting, and $\mathcal{R}(t)$ is more correlated to plasticity since the update on $\mathcal{R}(t)$ would have less impact on old tasks.

Experimental Evidence. We conduct a simple experiment to clearly show the rationality of the decoupling strategy. To be specific, we compare the mean interference on old knowledge caused by updates in $\mathcal{I}(t)$ and $\mathcal{R}(t)$. According to [18, 23, 31, 38], gradient updates within the feature space would do interference to old knowledge. Here we quantify this interference caused by the gradient $\boldsymbol{g}$ of a new task $t+1$ as

$$\omega(\boldsymbol{g}) = \sum_{j=1}^{t} \left\| \text{Proj}_{\mathcal{S}_j}(\boldsymbol{g}) \right\|_F^2 \tag{10}$$

where $\mathcal{S}_j$ is the feature subspace for task j. Obviously we have

$$\omega(\boldsymbol{g}) = \omega[\text{Proj}_{\bar{\mathcal{S}}(t)}(\boldsymbol{g})] = \omega(\boldsymbol{g}^{\mathcal{I}}) + \omega(\boldsymbol{g}^{\mathcal{R}}) \tag{11}$$

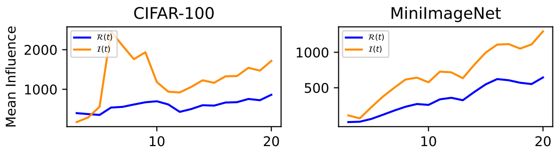

Figure 5. Mean knowledge interference of $\mathcal{I}(t)$ and $\mathcal{R}(t)$.

where $\boldsymbol{g}^{\mathcal{I}} = \text{Proj}_{\mathcal{I}(t)}(\boldsymbol{g})$ and $\boldsymbol{g}^{\mathcal{R}} = \text{Proj}_{\mathcal{R}(t)}(\boldsymbol{g})$. As a result, the mean knowledge interference of $\mathcal{I}(t)$ and $\mathcal{R}(t)$ can be defined as

$$\Omega(\mathcal{I}(t)) = \frac{\sum\limits_i \omega(\boldsymbol{g}_i^{\mathcal{I}})}{dim(\mathcal{I}(t))} \tag{12}$$

and

$$\Omega(\mathcal{R}(t)) = \frac{\sum\limits_i \omega(\boldsymbol{g}_i^{\mathcal{R}})}{dim(\mathcal{R}(t))} \tag{13}$$

where dim$(\cdot)$ is the dimension of space and i represents the index of data point in task $t+1$. $\Omega(\cdot)$ could eliminate the influence of the dimension of subspace and reflect the degree of interference on old tasks.

By calculating $\Omega(\mathcal{I}(t))$ and $\Omega(\mathcal{R}(t))$ on CIFAR100 and MiniImageNet under 20-split setting, we obtain results shown in Figure 5. We can easily find that gradient updates within subspace $\mathcal{I}(t)$ do significantly more interference on old tasks in the dimensional-average sense, which supports our claim. As a result, when leveraging feature space to balance stability and plasticity in continual learning, more stability should be taken into consideration for $\mathcal{I}(t)$, while more plasticity can be reserved for $\mathcal{R}(t)$.

3.3. ϵ and ζ on Stability / Plasticity Decoupling

Motivated by our decoupling strategy, we propose to differentially approximate these two kinds of feature subspaces. Shown in the right of Figure 3, for $\mathcal{I}(t)$ more bases should be reserved for stability and the gradient constraint should be stricter, thus larger ϵ and smaller ζ should be leveraged; while for $\mathcal{R}(t)$ more plasticity can be reserved, thus smaller ϵ and larger ζ is adopted. By performing such a refined balancing, we achieve higher accuracy and lower forgetting. Next, we give the detailed implementations.

$\mathcal{I}$ / $\mathcal{R}$ Construction. Obtaining $\mathcal{R}(t)$ and $\mathcal{I}(t)$ remains a challenging problem. To achieve this, we first establish the whole feature space *i.e.*, $\bar{\mathcal{S}}(t) = span\{\bar{\boldsymbol{S}}(t)\}$ for the first t tasks as Eq.(6) suggested. After constructing the feature matrix $\boldsymbol{M}_t$ of task t by Eq.(4), we first orthogonalize $\boldsymbol{M}_t$ with SVD to obtain the task-specific feature subspace $\mathcal{S}_t$. Then to find the shared feature spaces, we use a recurrent formulation:

$$\mathcal{I}(t) = \mathcal{I}(t-1) + \mathcal{F} \tag{14}$$

where $\mathcal{F} = \bar{\mathcal{S}}(t-1) \cap \mathcal{S}_t$ is the intersection between the historic feature space and current feature space. After that, the historic feature space can be updated as

$$\bar{\mathcal{S}}(t) = \bar{\mathcal{S}}(t-1) + \mathcal{S}_t \tag{15}$$

In this way, $\mathcal{I}(t)$ is updated task by task. In our implementation, we maintain these two subspaces *i.e.,* $\bar{\mathcal{S}}(t)$ and $\mathcal{I}(t)$ for efficient training.

After that, the orthogonal complement space $\mathcal{R}(t) = span\{\boldsymbol{R}(t)\}$ can be calculated by:

$$\boldsymbol{R}(t) = \bar{\boldsymbol{S}}(t) - \text{Proj}_{\mathcal{I}(t)}(\bar{\boldsymbol{S}}(t)). \tag{16}$$

In fact, we abuse the notation of $\boldsymbol{R}(t)$ here because it may contain duplicate bases, but we have noticed this simplification would not influence the subsequent procedures.

$\mathcal{I}$ / $\mathcal{R}$ Approximation. Based on previous discussions, we approximate $\boldsymbol{I}(t)$ and $\boldsymbol{R}(t)$ with different strength $\epsilon^{\mathcal{I}}$ and $\epsilon^{\mathcal{R}}$ to obtain the decoupled feature space $\hat{\mathcal{I}}(t) = span\{\hat{\boldsymbol{I}}(t)\}$ and $\hat{\mathcal{R}}(t) = span\{\hat{\boldsymbol{R}}(t)\}$, *i.e.*,

$$\hat{\boldsymbol{I}}(t) = \mathcal{A}(\boldsymbol{I}(t); \epsilon^{\mathcal{I}}) \tag{17}$$

and

$$\hat{\boldsymbol{R}}(t) = \mathcal{A}(\boldsymbol{R}(t); \epsilon^{\mathcal{R}}) \tag{18}$$

where $\epsilon^{\mathcal{I}}, \epsilon^{\mathcal{R}}$ are hyper-parameters predefined, and $\epsilon^{\mathcal{I}} > \epsilon^{\mathcal{R}}$. Here $\mathcal{A}$ is the approximation strategy defined in Section 3.1. Note that here we still have $\hat{\boldsymbol{I}}(t) \perp \hat{\boldsymbol{R}}(t)$, since $\hat{\boldsymbol{I}}(t) \subset \boldsymbol{I}(t)$ and $\hat{\boldsymbol{R}}(t) \subset \boldsymbol{R}(t)$. As a result, the approximated feature space $\hat{\mathcal{S}} = span\{[\hat{\boldsymbol{I}}(t), \hat{\boldsymbol{R}}(t)]\}$ is the actual feature space leveraged to constrain the gradient update.

$\mathcal{I}$ / $\mathcal{R}$ Projection. Another critical hyper-parameter ζ also significantly influences the performance. Similar to ϵ, for the intersection space $\hat{\mathcal{I}}(t)$ and the residual space $\hat{\mathcal{R}}(t)$, we put distinguishing constraints on the degree of orthogonality between new gradients and the feature space. Thus, the constraint function is modified as:

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{t+1} = \mathcal{C}(\nabla_{\boldsymbol{\theta}}\mathcal{L}_{t+1}; \{(\hat{\mathcal{I}}(t), \zeta^{\mathcal{I}}), (\hat{\mathcal{R}}(t), \zeta^{\mathcal{R}})\}) \tag{19}$$

where $\zeta^{\mathcal{I}}, \zeta^{\mathcal{R}}$ are hyper-parameters predefined, and $\zeta^{\mathcal{I}} < \zeta^{\mathcal{R}}$. Specifically, when projecting new gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{t+1}$ onto the orthogonal direction of $\hat{\mathcal{S}}(t)$ using Eq.(1), the constraint is modified as

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{t+1} =& \nabla_{\boldsymbol{\theta}}\mathcal{L}_{t+1} \\ &- \nabla_{\boldsymbol{\theta}}\mathcal{L}_{t+1}(1-\zeta^{\mathcal{I}})\hat{\boldsymbol{I}}(t)(\hat{\boldsymbol{I}}(t))^T \\ &- \nabla_{\boldsymbol{\theta}}\mathcal{L}_{t+1}(1-\zeta^{\mathcal{R}})\hat{\boldsymbol{R}}(t)(\hat{\boldsymbol{R}}(t))^T\end{aligned} \tag{20}$$

Note that here $[(1-\zeta^{\mathcal{I}})\hat{\boldsymbol{I}}(t)\hat{\boldsymbol{I}}(t)^T + (1-\zeta^{\mathcal{R}})\hat{\boldsymbol{R}}(t)\hat{\boldsymbol{R}}(t)^T]$ can be calculated before the training of every task, thus no extra time consumption is introduced to the optimizing stage compared to former methods [18, 23, 31, 38]. The above training procedures are summarized in Algorithm 3.

Algorithm 3: Feature Space Decoupling

```
Input: Datasets $\{\mathbb{D}_t\}$ for task $t \in \{t\}_{t=1}^{T}$, network $f$ parameterized by
        $\mathbb{W} = \{\theta^l\}_{l=1}^{L}$
Initialize $\bar{\mathcal{S}} = \emptyset, \mathcal{I} = \emptyset$
for task $t \in \{t\}_{t=1}^{T}$ do
    if $t > 1$ then
        Construct $\boldsymbol{M}_{t-1}$ using Eq.(4)
        Compute $\mathcal{S}_{t-1}$ using Eq.(5)
        Compute $\mathcal{F} = \bar{\mathcal{S}} \cap \mathcal{S}_{t-1}$ using Algorithm 1
        Update $\mathcal{I} = \mathcal{I} + \mathcal{F}$ using Algorithm 2
        Compute $\mathcal{R}$ using Eq.(16)
        Compute $\hat{\mathcal{I}}$ and $\hat{\mathcal{R}}$ using Eq.(17) and Eq.(18)
        update $\bar{\mathcal{S}} = \bar{\mathcal{S}} + \mathcal{S}_{t-1}$ using Algorithm 2
    while not converged do
        Sample a batch $\{\boldsymbol{x}, \boldsymbol{y}\}$ from $\mathbb{D}_t$
        Compute $f(\mathbb{W}; \{\boldsymbol{x}, \boldsymbol{y}\})$ and get backward gradient $\boldsymbol{g}$
        Compute restricted gradient $\hat{\boldsymbol{g}}$ using Eq.(20)
        Update $\mathbb{W}$ with $\hat{\boldsymbol{g}}$
```

4. Experiments

In this section, we conduct extensive experiments to verify the effectiveness of the proposed method. We apply our Space Decoupling algorithm (SD) to representative feature space based methods, such as GPM [31], TRGP [23] and Adam-NSCL [38] and show our method is model-agnostic and effective in balancing stability and plasticity. Next, we will introduce our experimental setups, main results, and ablation studies and show some visualization results.

4.1. Experimental Setting

Datasets. We evaluate our approach on CIFAR100 [19] and MiniImageNet [37]. **CIFAR100** is labeled as a subset of 80 million tiny image datasets. It contains 60,000 RGB images over 100 classes, with 500 images per class for training and 100 images per class for testing. Each image has a size of 32 × 32. **MiniImageNet** [37] is a 100-class subset of the original ImageNet [8] dataset. Each class contains 500 training images and 100 test images. the images are in RGB format of the size 84 × 84. Under continual learning setting, CIFAR100 and MiniImageNet are split into 10-split-CIFAR100, 20-split-CIFAR100 and 20-split-MiniImageNet, where the dataset is divided into 10, 20 and 20 tasks, respectively.

Implementation Details. Our SD algorithm is applied to GPM [31], Adam-NSCL [38] and TRGP [23]. All the implementations are consistent with the original paper during training and evaluation in our experiments. For example, when applying SD to GPM, we use a 5-layer AlexNet [20] as the backbone for 10-split-CIFAR-100 and 20-split-CIFAR-100, and ResNet18 [12] for 20-split-MiniImageNet. The initial learning rate is 0.01 for 10-split-CIFAR-100 and 20-split-CIFAR-100, and 0.1 for MiniImageNet. The batch size is set to 64 for all datasets. We perform grid-search on a validation set obtained by sampling 5% from the training set, and set $\epsilon^{\mathcal{I}} = 0.99, \epsilon^{\mathcal{R}} = 0.94$ in $\mathcal{I}/\mathcal{R}$ Approximation and $\zeta^{\mathcal{I}} = 1e-6, \zeta^{\mathcal{R}} = 5e-5$ in $\mathcal{I}/\mathcal{R}$ Projection. We compare SD with the baselines over 5 runs. Other details are presented in supplementary materials.

Compared Methods. We compare our approach with various representative continual learning methods, including LWF [22], EWC [17], MAS [2], MUC-MAS [24], GEM [25], A-GEM [5], AdNS [18] and OWM [41]. All of them are published in recent years and are relevant to our work. LWF leverages knowledge distillation [13] to preserve learned knowledge of previous tasks. EWC, MAS and MUC-MAS store important weight in memory. GEM, A-GEM and OWM focus on designing network training algorithms to overcome forgetting. AdNS leverages feature null space to guide the direction of gradient updates.

Evaluation Metrics. We employ average accuracy (ACC) and backward transfer (BWT) as our evaluation metrics, which are proposed in [25]. ACC is the average accuracy on the test dataset of all seen tasks, and BWT is the average drop in the accuracy of the network for the test dataset of previous tasks after learning the current task. Models with higer ACC and BWT are better.

4.2. Main Results

Table 1 shows the main results of our method compared with other SOTA approaches in terms of ACC and BWT. It appears that Space Decoupling (SD) can effectively improve the performance on all three datasets. On 20-split-CIFAR-100, GPM+SD, TRGP+SD and Adam-NSCL+SD improves ACC by 3.18%, 3.16% and 0.69% respectively; While on 10-split-CIFAR-100, GPM+SD, TRGP+SD and Adam-NSCL+SD improves ACC by 1.05%, 1.04% and 1.00% respectively. On 20-split-MiniImageNet, GPM+SD, TRGP+SD and Adam-NSCL+SD improves ACC by 1.98%, 2.29% and 1.31% respectively.

Besides, we can see the application of SD will not bring any extra forgetting in terms of BWT compared with the original methods. On 20-split-CIFAR-100 and 20-split-MiniImageNet, our approach even evidently improves this metric for GPM and TRGP. The reason might be our decoupling strategy divides the feature space into two subspaces, where different constraints on them enable us to find a better balance between stability and plasticity.

On 20-split-MiniImageNet, TRGP+SD achieves the highest ACC and BWT, which arrives at 65.8% and -0.49% respectively. The performance surpasses other methods by a large margin. On 20-split-CIFAR-100, TRGP+SD also achieves the highest ACC and BWT. All other methods like MAS and LWF fail to achieve comparable results as our TRGP+SD. On 10-split-CIFAR-100, Adam-NSCL+SD achieves the second-best ACC, a slight 1.24% lower than AdNS. However, AdNS requires to train another copy of the network, which results in tremendous

Model	Venue	20-split-MiniImageNet		20-split-CIFAR-100		10-split-CIFAR-100	
		ACC(%)	BWT(%)	ACC(%)	BWT(%)	ACC(%)	BWT(%)
LWF [22]	PAMI'17	57.63	-8.72	74.38	-9.11	70.7	-6.27
EWC [17]	PANS'17	52.01	-12	71.66	-3.72	70.77	-2.83
MAS [2]	ECCV'18	50.12	-5.82	63.84	-6.29	66.93	-4.03
MUC-MAS [24]	ECCV'20	46.24	-3.79	67.22	-5.72	63.73	-3.38
GEM [25]	NIPS'17	-	-	68.89	-1.2	49.48	2.77
A-GEM [5]	ICLR'18	57.24	-12	61.91	-6.88	49.57	-1.13
$*$AdNS [18]	ECCV'22	60.82	-4.24	77.33	-3.25	**77.21**	-2.32
OWM [41]	NMI'19	47.48	-8.57	68.47	-3.37	68.89	-1.88
GPM [31]	ICLR'21	60.41± 0.61	-0.7±0.4	77.53±0.83	-0.97±0.59	72.48±0.4	-0.9±0
Adam-NSCL [38]	CVPR'21	59.07±1.1	-4.9±1.32	75.81±0.93	-3.98±0.85	74.97±1.15	-2.64±0.91
TRGP [23]	ICLR'22	**63.51±0.74**	-0.76±0.25	**80.68±0.7**	-0.87±0.46	74.46±0.32	-0.9±0.01
GPM+SD		62.39±0.56	-0.61±0.11	80.71±0.82	-0.73±0.27	73.53±0.44	-0.83±0.31
Adam-NSCL+SD		60.38±0.75	-4.81±1	76.5±1.02	-3.99±0.96	**75.97±0.66**	-2.88±0.89
TRGP+SD		**65.8±0.16**	-0.49±0.08	**83.84±0.12**	-0.72±0.2	75.5±0.35	-0.96±0.09

Table 1. Quantitative results on various benchmark. Red and blue values denote the best and secondary performance. Methods that require to train another copy of the network is denoted by $*$.

Model	$\mathcal{I}/\mathcal{R}$-A	$\mathcal{I}/\mathcal{R}$-P	ACC(%)	BWT(%)
GPM			77.53±0.83	-0.97±0.59
	✓		79.99±0.48	-0.7±0.45
		✓	78.41±1.3	-1.44±0.65
	✓	✓	80.71±0.82	-0.73±0.27
TRGP			80.68±0.7	-0.87±0.46
	✓		83.21±0.36	-0.4±0.02
		✓	81.15±1.22	-1.24±0.71
	✓	✓	83.84±0.12	-0.72±0.2

Table 2. Ablation study of $\mathcal{I}/\mathcal{R}$ Approximation ($\mathcal{I}/\mathcal{R}$-A) and $\mathcal{I}/\mathcal{R}$ Projection ($\mathcal{I}/\mathcal{R}$-P) on 20-split-CIFAR-100.

memory and training time increase.

4.3. Ablation Studies

In this part, we will analyze the characteristic of our space decoupling strategy and demonstrate the effectiveness of each component in our approach.

Ablation on Approximation and Projection. As introduced in Section 3.3, we differently impose constraints on the feature subspaces $\mathcal{I}(t)$ and $\mathcal{R}(t)$, namely $\mathcal{I}/\mathcal{R}$ Approximation ($\mathcal{I}/\mathcal{R}$-A), and $\mathcal{I}/\mathcal{R}$ Projection ($\mathcal{I}/\mathcal{R}$-P). To verify their effectiveness, we perform experiments on 20-split-CIFAR-100 by selectively adding them to both GPM and TRGP. Table 2 shows the ablation results. It is observed that both $\mathcal{I}/\mathcal{R}$-A and $\mathcal{I}/\mathcal{R}$-P improve the ACC (*i.e.,* improves plasticity), due to the distinct constraints on $\mathcal{I}(t)$ and $\mathcal{R}(t)$. However, $\mathcal{I}/\mathcal{R}$-P would slightly degrade the BWT, indicating that a slight angle would slightly harm the stability but would remarkablely enhance the plasticity. In particular, by combining $\mathcal{I}/\mathcal{R}$-A and $\mathcal{I}/\mathcal{R}$-P together, we obtain a decent improvement in terms of ACC and BWT.

Ablation on the Feature Space Dimension. $\mathcal{I}/\mathcal{R}$ Approximation would reduce the dimension of feature space by selecting singular vectors using hyper-parameter ϵ. With different strengths, the "reduced" dimension mainly comes from subspace $\mathcal{R}$, which has less impact on old tasks. To verify this point, as is shown in Figure 6, the first row indicates the mean dimension (average the dimensions of layer-wise subspaces) expands as the incremental learning continues. Due to different ϵ, the dimension in SD increases much more slowly than GPM. The second and third rows show the changes of BWT and ACC, as the incremental tasks increase. It turns out that the mean accuracy of our approach is significantly increased due to the improved plasticity, while BWT keeps stable.

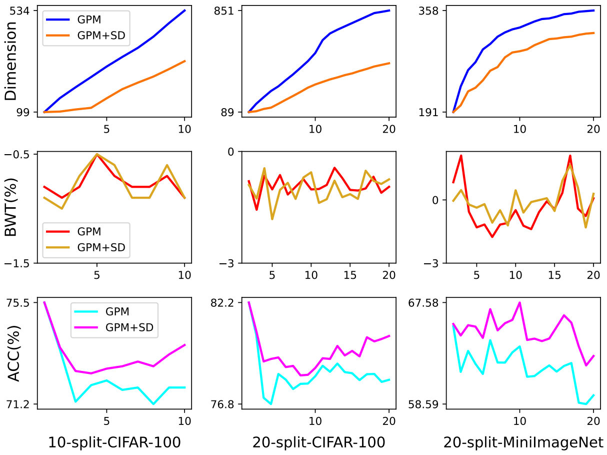

Figure 6. Comparison of feature space dimension, BWT and ACC of GPM and GPM+SD on different datasets. The feature space dimension is calculated by averaging the dimension of all layer-wise feature spaces.

We also perform an experiment to force the feature space of GPM to have the same dimension as SD. This is achieved by selecting k bases with the top-k largest singular values to reconstruct the subspace in GPM. The result is shown in the left of Figure 7. It seems that by applying the SD algorithm to GPM, ACC first decreases, and then improves

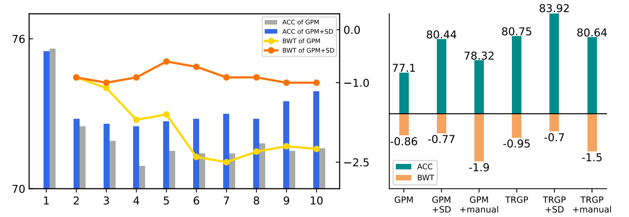

Figure 7. Left: Forcing GPM to have the same feature dimension with GPM+SD. Right: Comparisons with different subspaces construction. "manual" indicates manually forming $\mathcal{I}$ and $\mathcal{R}$ according to the singular value of bases. Experiments are conducted on 10-split-CIFAR-100 and 20-split-CIFAR-100 respectively.

as the mean dimension of the feature space decreases. On the contrary, BWT keeps stable by our approach while decreases sharply by GPM.

Ablation on the Subspace Construction. In Section 3.3, we propose to use Eq.(14) and Eq.(16) (subspace intersection) to establish $\boldsymbol{I}(t)$ and $\boldsymbol{R}(t)$. To illustrate its effectiveness, we perform an experiment to compare another construction strategy which also divides the feature space into two subspaces. Concretely, at the end of task t, we manually select k bases with the top-k largest singular values to form a subspace, where k is consistent with the dimension of $\boldsymbol{I}(t)$. This subspace is used to replace $\boldsymbol{I}(t)$ and $\boldsymbol{R}(t)$ is established in the same way. We say this construction strategy as manual.

The results on 20-split-CIFAR-100 are shown in Figure 7. As we can see, GPM suffers from severe forgetting while the performance of SD remains stable. That indicates our decoupling algorithm grasps the essence of stability and plasticity, and therefore yields better performance.

4.4. Model Analysis

Stability and Plasticity Analysis. Next, we explore the effect of $\{\epsilon^{\mathcal{I}}, \zeta^{\mathcal{I}}\}$ and $\{\epsilon^{\mathcal{R}}, \zeta^{\mathcal{R}}\}$ by varying their values. We perform experiments on 10-split-CIFAR-100 and present the results in Figure 8. We can draw two conclusions from the results. Firstly, the increase of ϵ or the decrease of ζcould improve the model's stability and lowers the plasticity; Secondly, the change of $\{\epsilon^{\mathcal{I}}, \zeta^{\mathcal{I}}\}$ results in tremendous fluctuation in model's stability compared to $\{\epsilon^{\mathcal{R}}, \zeta^{\mathcal{R}}\}$. For example, by fixing other hyper-parameters and tuning $\zeta^{\mathcal{R}}$ to $5e-4$, BWT falls to -1.34%. While the same change in $\zeta^{\mathcal{I}}$ results in a significantly larger drop to -2.88%. The above phenomenon supports our Space Decoupling (SD) theory that $\mathcal{I}$ has a higher correlation to old knowledge.

Computational Complexity. Memory: In terms of memory, the major difference is that Space Decoupling (SD) algorithm requires additional memory to store the intersection subspace $\mathcal{I}$. However, since the dimension of $\mathcal{I}$ would never overtake the dimension of the feature space, and is negligible compared to the memory of the network, the memory increase can be seen as zero.

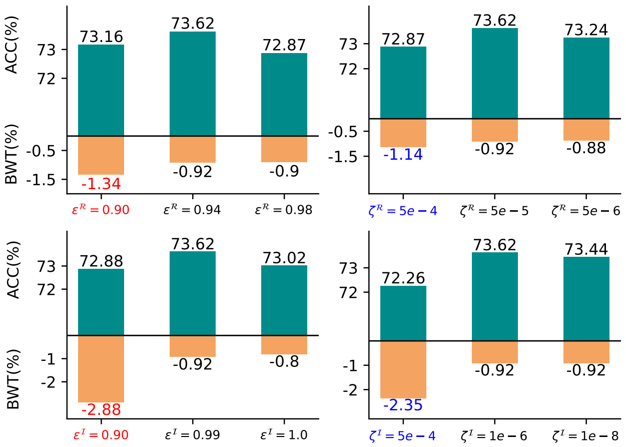

Figure 8. Stability and Plasticity analysis. Experiments are conducted by GPM+SD on 10-split-CIFAR-100. Red and blue values denote the comparison between $\mathcal{R}$ and $\mathcal{I}$.

Datasets	Methods					
	GPM	GPM+SD	TRGP	TRGP+SD	Adam-NSCL	Adam-NSCL+SD
10-split-CIFAR-100	0.25	0.27	0.41	0.45	2.71	2.93
20-split-CIFAR-100	0.31	0.36	0.48	0.53	4.14	4.53
20-split-MiniImageNet	0.58	0.66	0.81	0.92	11.28	12.95

Table 3. Comparison of training hours of different methods on all three datasets under the same environment.

Training time: We compare the training time of the above baselines before and after adding our SD algorithm under the same environment. Shown in Table 3, the training time increase is limited to a very small margin.

5. Conclusion

In this paper, we demonstrate the poor plasticity of recent gradient projection methods which is caused by constraining the gradient to be fully orthogonal to the whole feature space. Thus, we propose a Space Decoupling (SD) algorithm to decouple the feature space into stability-correlated space and plasticity-correlated space. By putting distinguishing constraints on the decoupled feature space, a better balance between stability and plasticity is achieved. Extensive experiments show that our proposed algorithm is model-agnostic and achieves SOTA performance on publicly available datasets.

Acknowledgments. This work is supported by grants from the National Key Research and Development Program of China (2021ZD0111000), National Natural Science Foundation of China (No.62222602, 62176092, 62106075), Natural Science Foundation of Shanghai (23ZR1420400), Shanghai Sailing Program (23YF1410500), Science and Technology Commission (No.21511100700), CAAI-Huawei MindSpore Open Fund, CCF-Lenovo Blue Ocean Research Fund.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018. 1

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018. 6, 7

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1

[4] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 1

[5] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018. 6, 7

[6] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. Continual learning with tiny episodic memories. 2019. 1

[7] Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020. 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[10] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146, 2019. 1

[11] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 1

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 6

[13] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 6

[14] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018. 1

[15] Ghassen Jerfel, Erin Grant, Tom Griffiths, and Katherine A Heller. Reconciling meta-learning and continual learning with online mixtures of tasks. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[16] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 6, 7

[18] Yajing Kong, Liu Liu, Zhen Wang, and Dacheng Tao. Balancing stability and plasticity through advanced null space in continual learning. *arXiv preprint arXiv:2207.12061*, 2022. 1, 2, 3, 4, 5, 6, 7

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 6

[21] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. *Advances in neural information processing systems*, 30, 2017. 1

[22] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 6, 7

[23] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Trgp: Trust region gradient projection for continual learning. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 4, 5, 6, 7

[24] Yu Liu, Sarah Parisot, Gregory Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *European Conference on Computer Vision*, pages 699–716. Springer, 2020. 6, 7

[25] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 6, 7

[26] 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. 1

[27] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013. 2

[28] Seyed Iman Mirzadeh, Mehrdad Farajtabar, and Hassan Ghasemzadeh. Dropout as an implicit gating mechanism for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 232–233, 2020. 2

[29] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of train-

ing regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020. 2

[30] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990. 1

[31] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2020. 1, 2, 3, 4, 5, 6, 7

[32] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016. 1

[33] Ghada Sokar, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Addressing the stability-plasticity dilemma via knowledge-aware continual learning. *arXiv preprint arXiv:2110.05329*, 2021. 2

[34] Concetto Spampinato, Simone Palazzo, Isaak Kavasidis, Daniela Giordano, Nasim Souly, and Mubarak Shah. Deep learning human mind for automated visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6809–6817, 2017. 1

[35] Gilbert Strang. *Linear algebra and its applications.* Belmont, CA: Thomson, Brooks/Cole, 2006. 2

[36] Jian Sun, Huibin Li, Zongben Xu, et al. Deep admm-net for compressive sensing mri. *Advances in neural information processing systems*, 29, 2016. 1

[37] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016. 6

[38] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 184–193, 2021. 1, 2, 3, 4, 5, 6, 7

[39] Zhen Wang, Liu Liu, Yiqun Duan, Yajing Kong, and Dacheng Tao. Continual learning with lifelong vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 171–181, 2022. 1

[40] Guile Wu, Shaogang Gong, and Pan Li. Striking a balance between stability and plasticity for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1124–1133, 2021. 2

[41] 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, 2019. 6, 7