Appendix: Balancing Stability and Plasticity through Advanced Null Space in Continual Learning

The content of the appendix is arranged as follows:

- **A Experiments**
  - **A.1 Implementation Details**
    In this part, we will introduce the low-rank approximation, details of $\alpha(t)$, k_l, and experimental setup.
  - **A.2 Additional Experimental Results**
    The additional experimental results include the comparison of running time, and exploration of plasticity and stability.
- **B Theoretical Analysis.** In this section, we provide the proof of Theorems 1 and 2, respectively.

A Experiments

This section provides the implementation details in Appendix A.1 and the additional experimental results in Appendix A.2.

A.1 Implementation Details

Low-Rank Approximation
Now we describe how to obtain $\mathbf{U}^l$ by solving the problem of low-rank approximation for the concatenation matrix $\tilde{\mathbf{U}}^l = [\mathbf{U}^l_{\text{pre}}, \mathbf{U}^l_{\text{cur}}] \in \mathbb{R}^{d^l \times n_0}$:

$$\begin{aligned} \text{minimize}_{\hat{\mathbf{U}}^l} \quad & \|\tilde{\mathbf{U}}^l - \hat{\mathbf{U}}^l\|_F \\ \text{s.t.} \quad & \text{Rank}(\hat{\mathbf{U}}^l) \leq k_l, l \in \{1, ..., L\}, \end{aligned} \tag{6}$$

where d^l is the dimension of the feature at l-th layer and n_0 is the sum of the colomns of $\mathbf{U}^l_{\text{pre}}$ and $\mathbf{U}^l_{\text{cur}}$. Let $\bar{\mathbf{U}}, \bar{\Sigma}, \bar{\mathbf{V}}^\top = \text{SVD}(\tilde{\mathbf{U}}^l)$, where $\bar{\Sigma}$ is a diagonal matrix sorted by singular values. Then the low-rank approximation matrix $\hat{\mathbf{U}}^l = \mathbf{U}^l \Sigma^l (\mathbf{V}^l)^\top$, where $\mathbf{U}^l \in \mathbb{R}^{d^l \times k_l}, \Sigma^l \in \mathbb{R}^{k_l \times k_l}, \mathbf{V}^l \in \mathbb{R}^{n_0 \times k_l}$; Σ^l is a diagonal matrix sorted by the k_l largest singular values, and $\mathbf{U}^l$ and $\mathbf{V}^l$ are constructed by the singular vectors corresponding to the k_l largest singular values in $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$, respectively. Then $\mathbf{U}^l$ is the objective matrix whose columns span the shared low-rank null space.
Details of $\alpha(t)$
Instead of solving the constraint $\|X^l_{t-1} \Delta w^l\|_1 \leq \epsilon(t)$, we use the function $\alpha(t)$ to replace $\epsilon(t)$ to balance the stability and plasticity. We propose non-uniform constraint strength, which linearly decreases with the task number t, i.e., $\alpha_t = \alpha_{\max} - \frac{t-1}{T-1}(\alpha_{\max} - \alpha_{\min})$, where $\alpha_{\max}$ and $\alpha_{\min}$ are the values of $\alpha(t)$ for the first and last task, respectively.
Details of k_l
When computing the shared low-rank null space, it is hard to set k_l for all

tasks and layers manually because we lack prior knowledge about the features at each layer. Therefore, we use a task-adaptive and layer-adaptive strategy to select k_l, i.e., we use the strategy of "Avg" in the paper to select k_l for all experiments. Specifically, assume that the dimensions of $\mathbf{U}_{\text{pre}}^l$ and $\mathbf{U}_{\text{cur}}^l$ are p and q, respectively. Then "Avg" means that $k = \text{Avg}(p, q) \times k_0$, where $\text{Avg}(\cdot)$ computes the average value of p and q, and k_0 is used to adjust the value of k. Note that we use the same operation for each layer. Therefore, we use k to denote k_l at each layer. Because the dimensions of the previous null space and the current candidate null space at each layer are determined by each task and the features at that layer, obtaining k by such strategy is task adaptive and layer-adaptive.

Datasets

The CIFAR100 dataset contains 100 classes, each of which has 500 training color images and 100 testing color images. TinyImageNet is a 200 classes dataset which contains 100,000 training images and 10,000 validation images and consists of 64×64 color images.

Experimental Setup

We use Pytorch[7] to implement the proposed algorithm and other experiments. The optimizer is Adam. Following [49], we use EWC [17] to regularize the parameters of batch normalization layer and set the regularization coefficient to 100. The learning rates for the first task are 2×10^{-4}, 1×10^{-4}, and 1×10^{-4} for 10-Split CIFAR-100, 20-Split CIFAR-100, and 25-Spilt TinyImageNet, respectively. Then the learning rate after the first task is set to 1×10^{-4}, 5×10^{-5}, and 1×10^{-4}, respectively. After that, we delay the learning rate at epoch 30 and 60 by multiplying with 0.5. The total epoch is 80 for all benchmarks. The batch size of 10-Split CIFAR-100, 20-Split CIFAR-100, and 25-Spilt TinyImageNet are 32, 16, and 16, respectively. We set $k_0 = 0.9$ for all benchmarks. The $\alpha_{\max}$ are 160, 180, and 20, and $\alpha_{\min}$ are 150, 150, and 5 for 10-Split CIFAR-100, 20-Split CIFAR-100, and 25-Spilt TinyImageNet, respectively.

A.2 Additional Experimental Results

Comparison of Running Time In this part, we compare the running time of the proposed method with Adam-NSCL [49], the most related baseline, to validate that the time consumption of the proposed method is comparable. The device is a single Nvidia Tesla V100 (16GB) GPU. As shown in Table 4, the running time of AdNS is comparable to Adam-NSCL on all benchmarks, validating that the time consumption of low-rank approximation is moderate.

Plasticity and Stability The effect of k. Like previous works [39], we use Learning Accuracy (LA), which is the accuracy of the model on a task right after it finishes training the task, to measure the plasticity. Higher LA indicates

[7] https://pytorch.org/

Table 4. Comparison of running time (s). The device is a single Nvidia Tesla V100 (16GB) GPU. The results of 10-Split CIFAR-100 and 20-Split CIFAR-100 are the average running time over five repetitions, and the results of 25-Split TinyImageNet are over three repetitions.

Method	**10-S-CIFAR-100**	**20-S-CIFAR-100**	**25-S-TinyImageNet**
Adam-NSCL [49]	5167 ± 5	23175 ± 416	32676 ± 1237
AdNS (Ours)	5623 ± 13	26286 ± 501	33462 ± 172

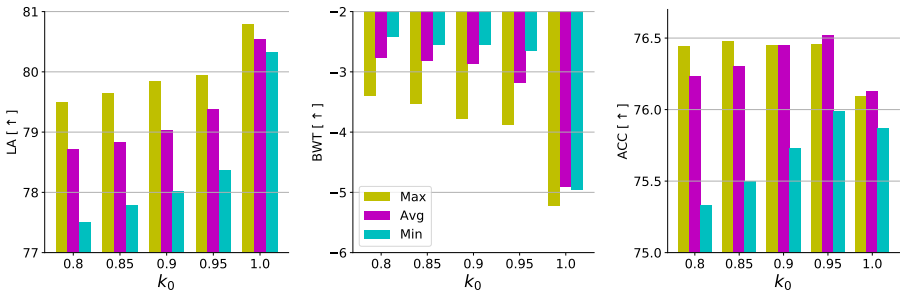

Fig. 5. The effect of k. Higher BWT indicates less forgetting (high stability) and higher LA indicates higher plasticity. The dataset is 10-Split CIFAR-100. [↑] higher is better.

better plasticity. In formal, LA is defined by

$$\text{LA} = \frac{1}{T}\sum_{i=1}^{T} A_{i,i}, \tag{7}$$

where T is the total number of tasks, and $A_{i,i}$ is the accuracy of task $\mathcal{T}_i$ after training on the task $\mathcal{T}_i$ sequentially. The larger the LA, the better the plasticity of the model. As shown in Fig. 5, for all strategies, with the increase of k_0, the plasticity becomes better while the forgetting becomes worse. The ACC first increases and then decreases due to the stability-plasticity trade-off. The results agree with the theoretical findings. With the increase of k_0, k becomes larger and the rank of the shared low-rank null space is larger. According to Theorems 1 and 2, it would result in better plasticity and worse forgetting. Finally, the performance of ACC will first increase and then decrease due to the stability-plasticity dilemma.

The effect of α. Larger α indicates looser constraint in (3), i.e., $\|X_{t-1}^l \Delta w^l\|_1 \leq \epsilon(t)$. As shown in Fig. 6, with the increase of α, LA becomes higher while BWT becomes worse. ACC first increases and then decreases as a result of the stability-plasticity dilemma.

The plasticity of different values of β. Now we validate that intra-task distillation is beneficial to improve the performance of the current task. Fig. 7

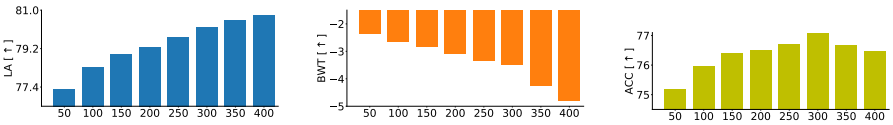

Fig. 6. The effect of α. Higher BWT indicates less forgetting (high stability) and higher LA indicates higher plasticity. The dataset is 10-Split CIFAR-100. [↑] higher is better.

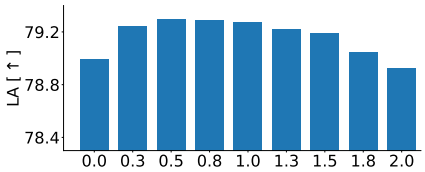

Fig. 7. The plasticity of different values of β. Higher LA indicates higher plasticity.

shows that with proper β, the LA can be improved, validating the effectiveness of intra-task distillation on improving the performance of the current task.

B Theoretical Analysis

In this part, we present the proof of Theorems 1 and 2, respectively.

Lemma 2. *Assume that $f(x)$ is L-smooth, let learning rate be $\eta \leq \frac{1}{L}$ and the update be $x_{t+1} = x_t - \eta \nabla h(x_t)$, we have*

$$\begin{aligned}\frac{\eta}{2}\|\nabla f(x_t)\|_2^2 \leq & f(x_t) - \mathbb{E}[f(x_{t+1})] + \frac{L\eta^2}{2}\mathbb{E}\|\nabla h(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2 \\ & + \frac{1}{2}\eta\|\nabla f(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2.\end{aligned}$$

Proof. Based on the definition of the smoothness of $f(x)$ and the update of $x_{t+1} = x_t - \eta \nabla h(x_t)$, we obtain,

$$\begin{aligned}f(x_{t+1}) = & f(x_t - \eta \nabla h(x_t)) \\ \leq & f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \frac{L}{2}\|x_{t+1} - x_t\|_2^2 \\ = & f(x_t) - \langle \nabla f(x_t), \eta \nabla h(x_t) \rangle + \frac{L}{2}\eta^2 \|\nabla h(x_t)\|_2^2.\end{aligned}$$

Taking expectation on both sides, we

$$\begin{aligned}\mathbb{E}[f(x_{t+1})] \leq & f(x_t) - \mathbb{E}\langle \nabla f(x_t), \eta \nabla h(x_t) \rangle + \frac{L}{2}\eta^2 \mathbb{E}\|\nabla h(x_t)\|_2^2 \\ = & f(x_t) - \eta \langle \nabla f(x_t), \mathbb{E}[\nabla h(x_t)] \rangle + \frac{L}{2}\eta^2 \mathbb{E}\|\nabla h(x_t) - \mathbb{E}[\nabla h(x_t)] + \mathbb{E}[\nabla h(x_t)]\|_2^2\end{aligned}$$

$$
\begin{aligned}
=&f(x_t) - \frac{1}{2}\eta \left(\|\nabla f(x_t)\|_2^2 + \|\mathbb{E}[\nabla h(x_t)]\|_2^2 - \|\nabla f(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2\right) \\
+&\frac{L\eta^2}{2}\left(\|\mathbb{E}[\nabla h(x_t)]\|_2^2 + \mathbb{E}\|\nabla h(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2 + 2\mathbb{E}\langle \mathbb{E}[\nabla h(x_t)], \nabla h(x_t) - \mathbb{E}[\nabla h(x_t)]\rangle\right) \\
=&f(x_t) - \frac{1}{2}\eta \left(\|\nabla f(x_t)\|_2^2 + \|\mathbb{E}[\nabla h(x_t)]\|_2^2 - \|\nabla f(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2\right) \\
&+ \frac{L}{2}\eta^2 \left(\|\mathbb{E}[\nabla h(x_t)]\|_2^2 + \mathbb{E}\|\nabla h(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2\right) \\
=&f(x_t) - \frac{1}{2}\eta\|\nabla f(x_t)\|_2^2 - \left(\frac{1}{2}\eta - \frac{1}{2}L\eta^2\right)\|\mathbb{E}[\nabla h(x_t)]\|_2^2 \\
&+ \frac{1}{2}L\eta^2\mathbb{E}\|\nabla h(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2 \\
&+ \frac{1}{2}\eta\|\nabla f(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2,
\end{aligned}
$$

where the second equation is based on the fact that $a^2 + b^2 - (a-b)^2 = 2ab$. By arranging, we have

$$
\begin{aligned}
\frac{1}{2}\eta\|\nabla f(x_t)\|_2^2 + \left(\frac{1}{2}\eta - \frac{1}{2}L\eta^2\right)\|\mathbb{E}[\nabla h(x_t)]\|_2^2 \leq& f(x_t) - \mathbb{E}[f(x_{t+1})] \\
&+ \frac{1}{2}L\eta^2\mathbb{E}\|\nabla h(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2 \\
&+ \frac{1}{2}\eta\|\nabla f(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2.
\end{aligned}
$$

If $\eta \leq \frac{1}{L}$, we have

$$
\begin{aligned}
\frac{1}{2}\eta\|\nabla f(x_t)\|_2^2 \leq& f(x_t) - \mathbb{E}[f(x_{t+1})] + \frac{1}{2}L\eta^2\mathbb{E}\|\nabla h(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2 \\
&+ \frac{1}{2}\eta\|\nabla f(x_t) - \mathbb{E}[\nabla h(x_t)]\|_2^2.
\end{aligned}
$$

□

Theorem. ***1*** *(Plasticity) Suppose Assumptions 1, 2, and 3 hold. Let $\mathbf{w}_{t,s}$ be the parameters on task $\mathcal{T}_t$ at the s-th step. Let the range of space of $\mathbf{U}^l$ be the null space of previous tasks for l-th layer, then the loss of the current task $\mathcal{T}_t$ after training on task $\mathcal{T}_t$ is upper bound by*

$$
\hat{L}_t(\mathbf{w}_{t,S}) \leq \hat{L}_t(\mathbf{w}_{t,0}) + \frac{\eta}{2}\sum_{s=0}^{S-1}\sum_{l=1}^{L}\|(I - \mathbf{U}^l(\mathbf{U}^l)^\top)g_{t,s}^l\|_2^2 - \frac{1}{2}\eta\sum_{s=0}^{S-1}\|\nabla\hat{L}_t(\mathbf{w}_{t,s})\|_2^2 + \frac{1}{2}SL_f\eta^2\sigma^2,
$$

where $g_{t,s}^l$ is l-th layer gradient of $\hat{L}_t(\mathbf{w}_{t,s})$.

Proof. When training on the task $\mathcal{T}_t$, based on Lemma 2, if $\eta \leq \frac{1}{L_f}$ we have

$$
\hat{L}_t(\mathbf{w}_{t,s+1}) \leq \hat{L}_t(\mathbf{w}_{t,s}) + \frac{1}{2}L_f\eta^2\sigma^2 - \frac{1}{2}\eta\|\nabla\hat{L}_t(\mathbf{w}_{t,s})\|_2^2 \tag{8}
$$

$$
+ \frac{1}{2}\eta\|\nabla\hat{L}_t(\mathbf{w}_{t,s}) - \left[(\mathbf{U}^1(\mathbf{U}^1)^\top(g_{t,s}^1)^\top \ldots (\mathbf{U}^L(\mathbf{U}^L)^\top(g_{t,s}^L)^\top\right]^\top\|_2^2 \tag{9}
$$

$$=\hat{L}_t(\mathbf{w}_{t,s}) + \frac{1}{2}L_f\eta^2\sigma^2 - \frac{1}{2}\eta\|\nabla\hat{L}_t(\mathbf{w}_{t,s})\|_2^2 + \frac{1}{2}\eta\sum_{l=1}^{L}\|(I-\mathbf{U}^l(\mathbf{U}^l)^\top)g_{t,s}^l\|_2^2. \tag{10}$$

Summing from $s=0$ to $s=S-1$, we obtain

$$\hat{L}_t(\mathbf{w}_{t,S}) \leq \hat{L}_t(\mathbf{w}_{t,0}) + \frac{\eta}{2}\sum_{s=0}^{S-1}\sum_{l=1}^{L}\|(I-\mathbf{U}^l(\mathbf{U}^l)^\top)g_{t,s}^l\|_2^2 - \frac{1}{2}\eta\sum_{s=0}^{S-1}\|\nabla\hat{L}_t(\mathbf{w}_{t,s})\|_2^2 + \frac{1}{2}SL_f\eta^2\sigma^2.$$

□

Theorem. *2 (Stability) Suppose Assumptions 1 and 2 hold. Let $\mathbf{w}_{t,s}$ be the parameters on task $\mathcal{T}_t$ at the s-th. Let $\hat{L}_{1:t-1}$ be the sum of empirical loss function of previous $t-1$ tasks and $g_{1:t-1,s}^l$ is its gradient of l-th layer at $\mathbf{w}_{t,s}$. Let $g_{t,s}^l$ be the gradient of the current task at $\mathbf{w}_{t,s}$ of l-th layer. Let the range of space of $\mathbf{U}^l$ be the null space of previous tasks for l-th layer, then the forgetting of previous $t-1$ tasks generated by the training on the task $\mathcal{T}_t$ is upper bound by*

$$\begin{aligned}\hat{L}_{1:t-1}(\mathbf{w}_{t,S}) - \hat{L}_{1:t-1}(\mathbf{w}_{t,0}) \leq & \eta\sum_{s=0}^{S-1}\sum_{l=1}^{L}\|\mathbf{U}^l(\mathbf{U}^l)^\top\|_2\|g_{t,s}^l\|_2\|g_{1:t-1,s}^l\|_2 \\ & + \frac{L_f}{2}\eta^2\sum_{s=0}^{S-1}\sum_{l=1}^{L}\|\mathbf{U}^l(\mathbf{U}^l)^\top\|_2^2\|g_{t,s}^l\|_2^2.\end{aligned}$$

Proof. Because the distributions between tasks are different, we could assume that $\langle\Delta\mathbf{w}_{t,s}, g_{1:t-1,s}\rangle \leq 0$. Based on the smoothness of $\hat{L}_{1:t-1}$, we obtain

$$\begin{aligned}\hat{L}_{1:t-1}(\mathbf{w}_{t,s+1}) &\leq \hat{L}_{1:t-1}(\mathbf{w}_{t,s}) - \eta\langle\Delta\mathbf{w}_{t,s}, \mathbf{g}_{1:t-1,s}\rangle + \frac{L_f}{2}\sum_{l=1}^{L}\|\eta\Delta\mathbf{w}_{t,s}\|_2^2 \\ &= \hat{L}_{1:t-1}(\mathbf{w}_{t,s}) - \eta\langle\Delta\mathbf{w}_{t,s}, \mathbf{g}_{1:t-1,s}\rangle + \frac{L_f}{2}\eta^2\sum_{l=1}^{L}\|\mathbf{U}^l(\mathbf{U}^l)^\top g_{t,s}^l\|_2^2 \\ &\leq \hat{L}_{1:t-1}(\mathbf{w}_{t,s}) - \eta\langle\Delta\mathbf{w}_{t,s}, \mathbf{g}_{1:t-1,s}\rangle + \frac{L_f}{2}\eta^2\sum_{l=1}^{L}\|\mathbf{U}^l(\mathbf{U}^l)^\top\|_2^2\|g_{t,s}^l\|_2^2,\end{aligned}$$

where $\mathbf{w}_{t,s+1} = \mathbf{w}_{t,s} - \eta\Delta\mathbf{w}_{t,s}$, $\mathbf{g}_{1:t-1,s}$ is the gradient of $\hat{L}_{1:t-1}$, and $g_{1:t-1,s}^l$ is the gradient of $\hat{L}_{1:t-1}$ at l-th layer.

Because of

$$\langle\Delta\mathbf{w}_{t,s}, \mathbf{g}_{1:t-1,s}\rangle = \sum_{l=1}^{L}\left\langle\mathbf{U}^l(\mathbf{U}^l)^\top g_{t,s}^l, g_{1:t-1,s}^l\right\rangle \leq \sum_{l=1}^{L}\|\mathbf{U}^l(\mathbf{U}^l)^\top\|_2\|g_{t,s}^l\|_2\|g_{1:t-1,s}^l\|_2,$$

we have

$$\hat{L}_{1:t-1}(\mathbf{w}_{t,s+1}) \leq \hat{L}_{1:t-1}(\mathbf{w}_{t,s}) + \eta\sum_{l=1}^{L}\|\mathbf{U}^l(\mathbf{U}^l)^\top\|_2\|g_{t,s}^l\|_2\|g_{1:t-1,s}^l\|_2$$

$$+ \frac{L_f}{2}\eta^2 \sum_{l=1}^{L} \|\mathbf{U}^l(\mathbf{U}^l)^\top\|_2^2 \|g_{t,s}^l\|_2^2.$$

Summing from $s = 0$ to $s = S - 1$, we obtain

$$\begin{aligned}\hat{L}_{1:t-1}(\mathbf{w}_{t,S}) - \hat{L}_{1:t-1}(\mathbf{w}_{t,0}) \leq & \eta \sum_{s=0}^{S-1} \sum_{l=1}^{L} \|\mathbf{U}^l(\mathbf{U}^l)^\top\|_2 \|g_{t,s}^l\|_2 \|g_{1:t-1,s}^l\|_2 \\ & + \frac{L_f}{2}\eta^2 \sum_{s=0}^{S-1} \sum_{l=1}^{L} \|\mathbf{U}^l(\mathbf{U}^l)^\top\|_2^2 \|g_{t,s}^l\|_2^2.\end{aligned}$$

□