Continuous Subspace Optimization for Continual Learning

Quan Cheng[1,2], **Yuanyu Wan**[3,4], **Lingyu Wu**[1,2], **Chenping Hou**[5], **Lijun Zhang**[1,2,*]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]School of Artificial Intelligence, Nanjing University, Nanjing, China
[3]School of Software Technology, Zhejiang University, Ningbo, China
[4]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security, Hangzhou, China
[5]College of Science, National University of Defense Technology, Changsha, China
{chengq, wuly, zhanglj}@lamda.nju.edu.cn
wanyy@zju.edu.cn, hcpnudt@hotmail.com

Abstract

Continual learning aims to learn multiple tasks sequentially while preserving prior knowledge, but faces the challenge of catastrophic forgetting when adapting to new tasks. Recently, approaches leveraging pre-trained models have gained increasing popularity in mitigating this issue, due to the strong generalization ability of foundation models. To adjust pre-trained models for new tasks, existing methods usually employ low-rank adaptation, which restricts parameter updates to a fixed low-rank subspace. However, constraining the optimization space inherently compromises the model's learning capacity, resulting in inferior performance. To address this limitation, we propose **Co**ntinuous **S**ubspace **O**ptimization for Continual Learning (CoSO) to fine-tune the model in a series of subspaces rather than a single one. These sequential subspaces are dynamically determined through the singular value decomposition of the gradients. CoSO updates the model by projecting gradients onto these subspaces, ensuring memory-efficient optimization. To mitigate forgetting, the optimization subspace of each task is constrained to be orthogonal to the historical task subspace. During task learning, CoSO maintains a task-specific component that captures the critical update directions for the current task. Upon completing a task, this component is used to update the historical task subspace, laying the groundwork for subsequent learning. Extensive experiments on multiple datasets demonstrate that CoSO significantly outperforms state-of-the-art methods, especially in challenging scenarios with long task sequences.

1 Introduction

Deep neural networks have achieved remarkable success when trained on large-scale offline data under the assumption of independent and identically distributed (i.i.d.) samples [He et al., 2016, Vaswani et al., 2017, Dosovitskiy et al., 2021]. However, real-world applications often require models to learn from a sequence of tasks with different data distributions, a scenario known as continual learning [De Lange et al., 2022, Van de Ven et al., 2022, Masana et al., 2022, Wang et al., 2024, Zhou et al., 2024b]. The major challenge in continual learning is catastrophic forgetting [McCloskey and Cohen, 1989], where the model's performance on previously learned tasks deteriorates significantly as it adapts to new tasks.

In recent years, pre-trained models especially vision transformers (ViTs) [Dosovitskiy et al., 2021] have demonstrated exceptional performance across various downstream tasks through their robust

*Lijun Zhang is the corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

generalization ability. This property makes pre-trained models highly promising in mitigating catastrophic forgetting, leading to a growing research focus on continual learning with foundation models [Smith et al., 2023, Lu et al., 2024, Liang and Li, 2024, Zhou et al., 2024a, Wu et al., 2025]. To efficiently fine-tune pre-trained ViTs, existing continual learning methods [Gao et al., 2023, Liang and Li, 2024, Wu et al., 2025] employ low-rank adaptation (LoRA) [Hu et al., 2022] to optimize the models, which confine parameter updates to a specific low-rank subspace to reduce the interference between tasks. However, this rigid constraint on update directions inherently limits the model's learning capacity, leading to inferior performance.

To address this issue, we propose **Co**ntinuous **S**ubspace **O**ptimization for Continual Learning (CoSO), which achieves enhanced adaptability by optimizing the model within multiple subspaces rather than a fixed one. These sequential subspaces are derived from the singular value decomposition of the gradients. By projecting gradients onto these low-dimensional subspaces for Adam [Kingma, 2014] optimization and then projecting back for parameter updates, CoSO achieves memory-efficient learning. To prevent forgetting, we enforce orthogonality between the optimization subspaces of current and historical tasks during training. While learning a task, CoSO leverages Frequent Directions (FD) [Ghashami et al., 2016, Wan and Zhang, 2018, 2022] to maintain a compact task-specific component, which captures critical update directions of the current task with negligible computational cost. After completing the current task, we use this dedicated component to estimate the task-specific subspace, which is then integrated into the historical task subspace, laying the groundwork for subsequent learning.

Experimental results on CIFAR100, ImageNet-R, and DomainNet show that CoSO consistently outperforms state-of-the-art methods by a significant margin across diverse continual learning settings, particularly in challenging scenarios involving long task sequences. The substantial performance gains highlight CoSO's strong potential for real-world continual learning.

In summary, our contributions are as follows:

- We propose CoSO, a novel continual learning framework that fine-tunes pre-trained models via continuous gradient-derived subspaces, enabling efficient adaptation to sequential tasks.
- We introduce a lightweight mechanism to maintain the historical task subspace, enabling CoSO to keep current updates orthogonal to the historical subspace and thereby mitigate task interference.
- We conduct extensive experiments, demonstrating CoSO's superior performance over prior PEFT-based continual learning methods across various datasets and settings.

2 Related Work

In this section, we review related work on continual learning and low-rank optimization in offline learning.

2.1 Continual Learning

Continual learning [De Lange et al., 2022, Van de Ven et al., 2022, Masana et al., 2022, Wang et al., 2024, Zhou et al., 2024b] aims to enable neural networks to incrementally learn from a sequence of tasks while retaining previously learned knowledge. These approaches broadly fall into five categories [Wang et al., 2024]: regularization-based methods [Zenke et al., 2017, Kirkpatrick et al., 2017, Li and Hoiem, 2017], replay-based methods [Lopez-Paz and Ranzato, 2017, Rebuffi et al., 2017, Chaudhry et al., 2019a,b, Liu et al., 2020, Sun et al., 2022], optimization-based methods [Farajtabar et al., 2020, Saha et al., 2021, Wang et al., 2021], representation-based methods [Madaan et al., 2022, Pham et al., 2024], and architecture-based methods [Yoon et al., 2018, Li et al., 2019, Sokar et al., 2021, Liang and Li, 2023]. Regularization-based methods introduce additional loss terms to constrain parameter updates, preventing drastic changes in parameters that are important for early tasks. Replay-based methods store a small subset of training samples from previous tasks in a limited buffer and periodically replay these samples alongside new data, allowing the model to rehearse earlier knowledge. Optimization-based methods manipulate the update directions of each task according to preserved information of previous tasks. Representation-based methods utilize statistical information of features to calibrate classifiers. Architecture-based methods dynamically modify network architectures, dedicating specific model capacity for new tasks.

Early continual learning approaches typically initialize their models with random weights. The strong generalization capabilities of foundation models, especially vision transformers [Dosovitskiy et al., 2021], have made pre-trained architectures an increasingly attractive solution for continual learning [Zhou et al., 2024a]. Recent developments in parameter-efficient fine-tuning (PEFT) based continual learning methods [Gao et al., 2023, Liang and Li, 2024, Lu et al., 2024, Wu et al., 2025] have facilitated efficient adaptation of foundation models through selective parameter optimization, substantially lowering computational requirements. Existing PEFT-based methods can be broadly categorized into two groups: (1) prompt-based techniques that focus on optimizing learnable tokens [Lester et al., 2021, Wang et al., 2022a,b, Smith et al., 2023, Lu et al., 2024], and (2) LoRA-based methods that adjust parameters within constrained low-dimensional subspaces [Gao et al., 2023, Liang and Li, 2024, Wu et al., 2025].

Among prompt-based approaches, L2P [Wang et al., 2022b] introduces task-specific prompt tokens to modulate the pre-trained model's behavior, but struggles with knowledge transfer between tasks. DualPrompt [Wang et al., 2022a] addresses this limitation by maintaining both task-specific and task-invariant prompts, enabling better knowledge sharing. CODA-Prompt [Smith et al., 2023] further enhances adaptation flexibility through dynamic prompt composition from a shared pool. VPT-NSP^2 [Lu et al., 2024] learns each task by tuning learnable prompts in the null space of previous tasks' features. However, these methods influence model behavior indirectly through learnable tokens, which restrict the model's ability to capture complex task-specific features.

Complementary to prompt-based methods, LoRA-based approaches directly update model parameters in a parameter-efficient manner. InfLoRA [Liang and Li, 2024] constrains the parameter updates within a predetermined subspace to reduce the interference between tasks. SD-LoRA [Wu et al., 2025] decouples the learning of the magnitude and direction of LoRA components. However, both methods confine weight updates to a specific low-rank subspace, which inherently limits the model's learning capacity. Unlike these methods, CoSO updates the parameters across a series of subspaces, enabling the learning of full-rank weights and thereby enhancing the model's flexibility.

2.2 Low-rank Optimization in Offline Learning

Low-rank adaptation (LoRA) [Hu et al., 2022] has gained significant attention for its ability to reduce computational and memory requirements when fine-tuning pre-trained models [Mao et al., 2022, Zhang et al., 2023]. Specifically, LoRA reparameterizes the update of a linear layer's weights $\Delta W = BA \in \mathbb{R}^{m\times n}$, where $B \in \mathbb{R}^{m\times r}, A \in \mathbb{R}^{r\times n}$ are low-rank matrices. By freezing the original weights and only updating the low-rank components, LoRA enables parameter-efficient fine-tuning while preserving performance in many downstream tasks. However, it has been demonstrated [Xia et al., 2024] that low-rank weight updates limit the performance compared to full-rank fine-tuning. Recent works [Cosson et al., 2023, Zhao et al., 2024] have shown that neural network gradients often exhibit low-rank structure. Instead of approximating the weight matrix as low rank, GaLore [Zhao et al., 2024] directly leverages the low-rank gradients to optimize the model. This methodology enables memory-efficient optimization through effective dimensionality reduction in gradient spaces.

To be concrete, GaLore utilizes the singular value decomposition (SVD) of $G_t \in \mathbb{R}^{m\times n}$ to compute a low-rank projection matrix $P_t \in \mathbb{R}^{m\times r}$, where $r \ll n$ is the target rank. Leveraging P_t, GaLore transforms the gradient G_t into a compact form $P_t^\top G_t$ to achieve memory-efficient parameter updates. At each training step t, the gradient update can be decomposed into three operations:

$$\begin{aligned} R_t &= P_t^\top G_t && \text{(forward projection)} \\ N_t &= \text{Adam}(R_t) && \text{(adam optimizer update)} \\ \tilde{G}_t &= P_t N_t. && \text{(backward projection)} \end{aligned}$$

The projection matrix P_t is periodically updated through SVD to follow the evolving gradient subspace. Utilizing the final gradient $\tilde{G}_t$, GaLore updates the model parameters with learning rate η:

$$W_t = W_{t-1} - \eta \tilde{G}_t.$$

Compared to LoRA, GaLore not only reduces memory storage from $(mn + 3mr + 3nr)$ to $(mn + mr + 2nr)$, but also achieves higher model capacity by directly optimizing in the most relevant gradient subspaces rather than constraining updates to a predefined low-rank structure.

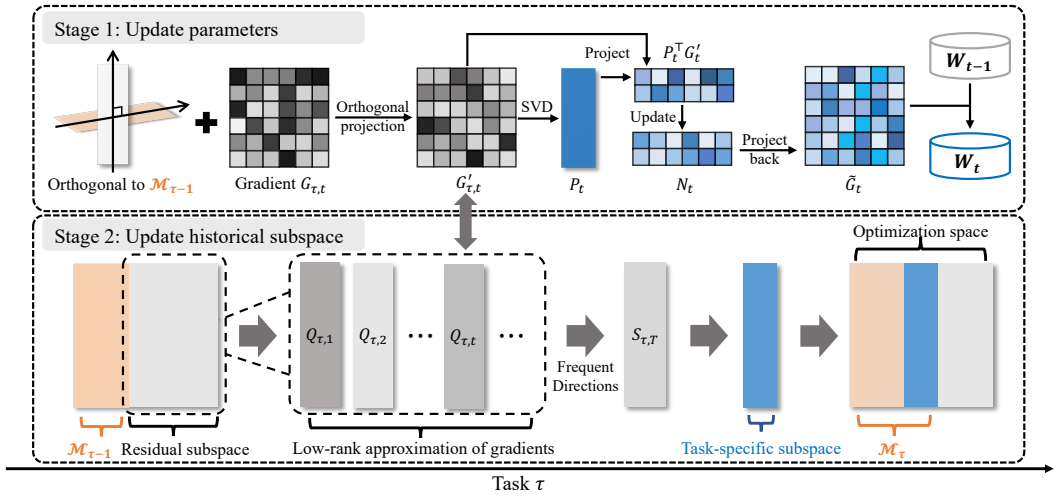


Figure 1: CoSO optimizes the parameters in continual low-rank subspaces, enhancing the learning capacity of models. To mitigate forgetting, the optimization subspaces of the current task are set to be orthogonal to the historical task subspace. While learning a task, CoSO consolidates the low-rank approximation matrices $\{Q_{\tau,t}\}_{t=1}^{T}$ into a task-specific component $S_{\tau,T}$ through Frequent Directions. The dedicated component is then used to update the historical task subspace spanned by $\mathcal{M}_{\tau-1}$.

3 Methodology

In this section, we first introduce the necessary preliminaries, then present the details of our approach.

3.1 Preliminaries

In continual learning, a model needs to learn a sequence of tasks while retaining knowledge of previous tasks. We consider the class-incremental learning setting, where task identities are unavailable at inference time and access to historical data is prohibited during learning new tasks [Wang et al., 2022a, Smith et al., 2023, Liang and Li, 2024, Lu et al., 2024, Wu et al., 2025]. We denote the task sequence as $\mathcal{D} = \{\mathcal{D}_1, ..., \mathcal{D}_N\}$, where each task dataset $\mathcal{D}_\tau = \{(\mathbf{x}_{i,\tau}, y_{i,\tau})\}_{i=1}^{n_\tau}$ contains n_τ input-label pairs. Following recent work [Wang et al., 2022a, Gao et al., 2023], we adopt a pre-trained Vision Transformer (ViT) [Dosovitskiy et al., 2021] as the backbone network, denoted as $f_\Theta(\cdot)$ with parameters Θ, and classifier $h_\Phi(\cdot)$ with parameters Φ, thus the model is $h_\Phi(f_\Theta(\cdot))$. Formally, the hidden state Y_τ^ℓ of feature X_τ^ℓ at the linear layer ℓ, can be calculated as $Y_\tau^\ell = W^\ell X_\tau^\ell$, where W^ℓ is the weight matrix of the linear layer. Let $G_{\tau,t}^\ell$ denote the gradient at the t-th training step of the linear layer ℓ in task τ. For simplicity, we omit the symbol ℓ, using W to refer W^ℓ and $G_{\tau,t}$ to refer $G_{\tau,t}^\ell$ in the following sections.

3.2 Continuous Subspaces Optimization

Inspired by GaLore [Zhao et al., 2024], we propose CoSO to address the rigidity of single subspace adaptation methods through multiple subspaces optimizing. However, directly using GaLore causes severe interference between different tasks in continual learning. To minimize the interference, CoSO enforces orthogonal constraints between current and historical subspace during training. Motivated by memory consolidation in cognitive neuroscience [Dudai, 2004], CoSO estimates a task-specific subspace to consolidate knowledge upon learning each task. This subspace preserves critical learning directions of the task based on gradients at all training steps, and is incrementally integrated into the historical task subspace, enabling efficient knowledge accumulation. The whole process of CoSO is illustrated in Figure 1. We first introduce how to optimize the model in continuous subspaces in this section. Then we present how to update the historical task subspace in Section 3.3.

In continual learning, the key challenge is to prevent new task updates from interfering with previously learned knowledge. Building on the insight that gradient updates in neural networks typically lie in the span of input features [Saha et al., 2021], we develop an approach that leverages this gradient-input feature relationship to minimize task interference through orthogonal projection. Specifically, we maintain an orthogonal basis matrix $\mathcal{M}_{\tau-1}$ that spans the gradient subspace accumulated from all previous tasks prior to the current task τ. Since gradients inherently encode information about the input features they were computed from, this historical subspace captures the principal directions that were important for learning previous tasks. We recognize that gradient steps along these historical directions would cause maximal interference with past learning, while gradient steps orthogonal to this space result in minimal interference. For each gradient $G_{\tau,t}$ computed during training on task τ, we project it onto the orthogonal complement of historical subspace:

$$G'_{\tau,t} = G_{\tau,t} - \mathcal{M}_{\tau-1}\mathcal{M}_{\tau-1}^{\top}G_{\tau,t}. \tag{1}$$

This projection removes the gradient component aligned with the learning directions of previous tasks, leaving only the orthogonal component $G'_{\tau,t}$ for updating the parameters. When we update the weight matrix using these orthogonal gradients, i.e., $\Delta W = -\eta \sum_{t=1}^{T} G'_{\tau,t}$, the parameter changes occur in directions that have minimal overlap with the optimization trajectories of previous tasks. This approach effectively partitions the parameter space, preserving directions important for past tasks while utilizing orthogonal directions for new learning. The orthogonal projection thus provides a principled way to balance plasticity for new tasks with stability for old tasks, enabling the model to expand its capabilities while mitigating interference.

However, updating the model with the full orthogonal gradient $G'_{\tau,t}$ incurs substantial memory overhead and high computational cost, particularly in vision transformers. To achieve memory-efficient fine-tuning, we follow GaLore [Zhao et al., 2024] and decompose $G'_{\tau,t} \in \mathbb{R}^{m\times n}$ using singular value decomposition (SVD) to get the projection matrix $P_{\tau,t}$:

$$\begin{aligned} U\Sigma V^{\top} &= \text{SVD}_{r_1}(G'_{\tau,t}) \\ P_{\tau,t} &= U[:, : r_1], \end{aligned} \tag{2}$$

where m and n are the dimensions of the original weight matrix, $r_1 \ll n$ is the target projection rank, and $\text{SVD}_{r_1}(\cdot)$ denotes a truncated SVD that retains the top-r_1 singular values. Subsequently, we project the orthogonalized gradient $G'_{\tau,t}$ into the low-rank subspace spanned by $P_{\tau,t}$, effectively reducing the memory footprint of parameter updates:

$$R_{\tau,t} = P_{\tau,t}^{\top}G'_{\tau,t}. \tag{3}$$

Then $R_{\tau,t}$ is updated by Adam [Kingma, 2014] as follows:

$$\begin{aligned} M_{\tau,t} &= (\beta_1 \cdot M_{\tau,t-1} + (1-\beta_1)\cdot R_{\tau,t}) \,/(1-\beta_1^t) \\ V_{\tau,t} &= \left(\beta_2 \cdot V_{\tau,t-1} + (1-\beta_2)\cdot R_{\tau,t}^2\right)/(1-\beta_2^t) \\ N_{\tau,t} &= M_{\tau,t}/\left(\sqrt{V_{\tau,t}} + \epsilon\right), \end{aligned} \tag{4}$$

where β_1, β_2 are decay rates, $M_{\tau,t}$ is the first-order momentum, and $V_{\tau,t}$ is the second-order momentum. The low-rank normalized gradient $N_{\tau,t}$ is then projected back to update the parameters with learning rate η:

$$\begin{aligned} \tilde{G}_{\tau,t} &= P_{\tau,t}N_{\tau,t} \\ W_{\tau,t} &= W_{\tau,t-1} - \eta \cdot \tilde{G}_{\tau,t}. \end{aligned} \tag{5}$$

Because $P_{\tau,t}$ is computed from the projected gradient $G'_{\tau,t}$, which is orthogonal to the historical subspace spanned by $\mathcal{M}_{\tau-1}$, any parameter updates derived from $P_{\tau,t}$ remain in the null space of previous tasks' feature spaces. Consequently, the linear layer's output for every earlier task remains unchanged, preventing interference at the representation level. Since $P_{\tau,t}$ is dynamically changed to capture the most important directions of $G'_{\tau,t}$, we are optimizing the model in continuous subspaces rather than a fixed one, thereby expanding the model's representational adaptability. To balance computational efficiency, we update the projection matrix $P_{\tau,t}$ every K steps. By updating $R_{\tau,t}$ in lower dimension space, the memory requirement is reduced from $(mn + 3mr_1 + 3nr_1)$ to $(mn + mr_1 + 2nr_1)$ compared to LoRA-based methods, such as InfLoRA [Liang and Li, 2024] and SD-LoRA [Wu et al., 2025].

3.3 Historical Task Space Update

Task-Specific Subspace Estimation. To update the orthogonal basis matrix $\mathcal{M}_{\tau-1}$ of the historical task space, we need to efficiently estimate a task-specific subspace, which retains the critical gradient directions of the current task. Specifically, for task τ, the model undergoes T training steps, producing a sequence of gradients $\{G'_{\tau,1}, ..., G'_{\tau,T}\}$, where $G'_{\tau,t} \in \mathbb{R}^{m\times n}$. To identify the primary directions of these gradients, we consider the accumulated covariance matrix $\sum_{t=1}^{T} G'_{\tau,t} G'^{\top}_{\tau,t} \in \mathbb{R}^{m\times m}$, which integrates information from all training steps and characterizes the subspace where most updates occur. However, directly maintaining such accumulated covariance matrix would be computationally expensive, requiring $O(m^2nT)$ time complexity. This is particularly challenging for transformer-based models where the parameter dimension m, n are typically in the order of thousands.

To ensure computational efficiency, we use Frequent Directions (FD) [Ghashami et al., 2016, Wan and Zhang, 2018, 2022], a deterministic matrix sketching algorithm, to maintain a low-rank approximation of streaming gradients. The FD algorithm processes the gradients sequentially while providing a guarantee on approximation quality [Wan and Zhang, 2021, Yang et al., 2025]. Specifically, we first compute a low-rank matrix $Q_{\tau,t} \in \mathbb{R}^{m\times r_2}$ with $r_2 \ll n$ through singular value decomposition (SVD):

$$\begin{aligned} U\Sigma V^{\top} &= \text{SVD}_{r_2}(G'_{\tau,t}) \\ Q_{\tau,t} &= U\Sigma. \end{aligned} \tag{6}$$

Here, $\text{SVD}_{r_2}(\cdot)$ denotes a truncated SVD that retains the top-r_2 singular values. The resulting low-rank matrix $Q_{\tau,t}$ enables us to efficiently approximate the gradient covariance matrix:

$$Q_{\tau,t} Q^{\top}_{\tau,t} \approx G'_{\tau,t} G'^{\top}_{\tau,t}. \tag{7}$$

Based on this approximation, we further compute a sketch matrix $S_{\tau,t} \in \mathbb{R}^{m\times r_2}$ that incrementally consolidates the gradient covariance information from all training steps up to step t. This consolidation is achieved by combining the previous sketch matrix $S_{\tau,t-1}$ with the current approximation $Q_{\tau,t}$. The update of $S_{\tau,t}$ is as follows:

$$\begin{aligned} U'\Sigma' V'^{\top} &= \text{SVD}_{r_2}([S_{\tau,t-1}, Q_{\tau,t}]) \\ S_{\tau,t} &= U'\sqrt{\Sigma'^2 - \sigma_t I_{r_2}}, \sigma = \Sigma'^2_{r_2,r_2}. \end{aligned} \tag{8}$$

We initialize $S_{\tau,1} = Q_{\tau,1}$, and after T iterations, we obtain the final task-specific sketch matrix $S_{\tau,T}$, which satisfies:

$$S_{\tau,T} S^{\top}_{\tau,T} \approx \sum_{t=1}^{T} G'_{\tau,t} G'^{\top}_{\tau,t}. \tag{9}$$

By analyzing the dominant singular vectors of $S_{\tau,T}$, we can effectively estimate the principal subspace of the current task. Note that we update $S_{\tau,t}$ every K steps to match the update frequency of projection matrix $P_{\tau,t}$, ensuring consistency in our approximation process. The effectiveness of CoSO relies on the accuracy of low-rank approximation, which is formalized through the following Proposition 1.

Proposition 1. *Given a sequence of projected gradients $\{G'_{\tau,t}\}_{t=1}^{T}$ and low-rank matrix $\{Q_{\tau,t}\}_{t=1}^{T}$, where $G'_{\tau,t} \in \mathbb{R}^{m\times n}$ and $Q_{\tau,t} \in \mathbb{R}^{m\times r_2}$. The final sketch matrix is $S_{\tau,T} \in \mathbb{R}^{m\times r_2}$. Let $A = \sum_{t=1}^{T} G'_{\tau,t} G'^{\top}_{\tau,t}$, $\tilde{A} = \sum_{t=1}^{T} Q_{\tau,t} Q^{\top}_{\tau,t}$. For any $k < r_2$ the approximation error is bounded by:*

$$\|A - S_{\tau,T} S^{\top}_{\tau,T}\|_2 \leq \sum_{t=1}^{T} \sigma_t^2 + \frac{\|\tilde{A} - [\tilde{A}]_k\|_F^2}{r_2 - k}, \tag{10}$$

where σ_t is the (r_2+1)-th singular value of $G'_{\tau,t}$ and $[\tilde{A}]_k$ is the minimizer of $\|\tilde{A} - [\tilde{A}]_k\|_F$ overall rank k matrices.

Remark. *Because the gradients often exhibit low-rank structure [Cosson et al., 2023, Zhao et al., 2024], their singular values decay rapidly. Consequently, the error $\sum_{t=1}^{T} \sigma_t^2$ would be negligibly small when r_2 exceeds the intrinsic rank of the gradients. By maintaining low-rank sketch matrix $S_{\tau,T}$, we reduce the cost of computing $\sum_{t=1}^{T} G'_{\tau,t} G'^{\top}_{\tau,t}$ from $O(m^2nT)$ to $O(mnr_2T)$, where $r_2 \ll m$. Proposition 1 ensure that our low-rank approximation captures the most significant directions in the gradient space. The error bound provides practical guidance for choosing the rank r_2: larger values lead to better approximation at the expense of additional computation and memory. To better preserve the task information, we set r_2 to be slightly larger than r_1, where r_1 is the projection rank introduced in Section 3.2. The proof is provided in Appendix A.*

Update Orthogonal Basis Matrix. Once the final task-specific sketch matrix $S_{\tau,T}$ is computed, we use it to update the orthogonal basis matrix $\mathcal{M}_{\tau-1}$ to incorporate the optimization subspace of the task τ. First, we extract the principal directions of the current task by performing SVD on its sketch matrix:

$$U_\tau \Sigma_\tau V_\tau^\top = \text{SVD}(S_{\tau,T}). \tag{11}$$

Then, we determine the number of directions to retain based on the sum of squared singular values. Following the principle of matrix approximation with SVD, we select k as the biggest value that satisfies:

$$\frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{j=1}^{r_2} \sigma_j^2} \leq \epsilon_{th}, \tag{12}$$

where $\epsilon_{th} \in (0, 1]$ is a threshold hyperparameter controlling the ratio to preserve, and σ_i is the i-th singular values in descending order. This criterion ensures that the selected k directions capture at least ϵ_{th} fraction of the total variance in the gradient space. Finally, we expand the orthogonal basis matrix $\mathcal{M}_{\tau-1}$ by incorporating these new directions:

$$\mathcal{M}_\tau = [\mathcal{M}_{\tau-1}, U_\tau[:, : k]]. \tag{13}$$

The above selection and update process ensure that we capture the most important learning directions for each task while maintaining orthogonality between different tasks' subspaces.

Due to space constraints, the complete CoSO algorithm is presented in Appendix B.

4 Experiments

We conduct comprehensive experiments with varying numbers of sequential tasks to evaluate CoSO's effectiveness across multiple datasets. We first outline our experimental settings, then present detailed results and analyses.

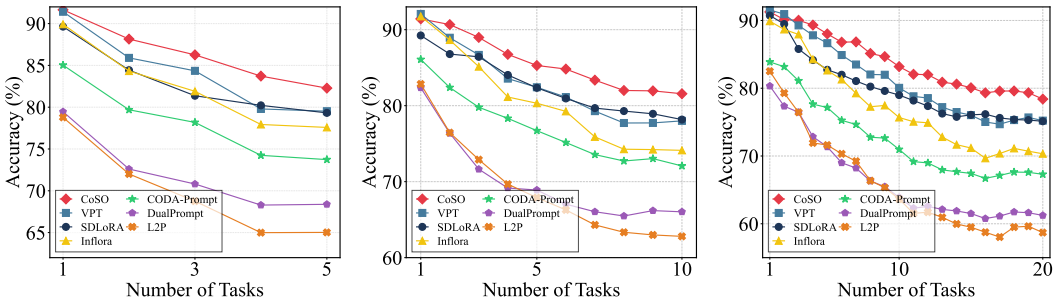
4.1 Experimental Settings

Datasets and Evaluation Metrics. Following previous works [Wang et al., 2022b, Liang and Li, 2024], we evaluate CoSO on three widely-used continual learning benchmarks: ImageNet-R [Hendrycks et al., 2021], CIFAR100 [Krizhevsky, 2009], and DomainNet [Peng et al., 2019]. ImageNet-R contains 200 classes from ImageNet with artistic style variations. Similar to existing works [Smith et al., 2023, Liang and Li, 2024, Wu et al., 2025], we create three different splits of ImageNet-R: 5 tasks with 40 classes per task, 10 tasks with 20 classes per task, and 20 tasks with 10 classes per task. For CIFAR100, we divide it into 10 tasks, each containing 10 classes. DomainNet consists of 345 classes across six distinct domains and is split into 5 tasks, with 69 classes per task.

We evaluate our method using two complementary metrics that are widely adopted in existing continual learning methods [Wang et al., 2022b, Liang and Li, 2024, Wu et al., 2025]. The first metric is the final accuracy ACC_T, which evaluates the model's overall performance across all tasks after the complete training process. The second metric is the average accuracy $\overline{ACC}_T$, which measures the model's learning stability throughout the training sequence and is calculated as $\overline{ACC}_T = \frac{1}{T} \sum_{i=1}^{T} ACC_i$, where T denotes the total number of tasks. These two metrics capture both the model's ability to learn new tasks and retain knowledge of previously learned tasks, providing a comprehensive assessment of continual learning performance.

Table 1: Results (%) on ImageNet-R with varying numbers of tasks (5, 10 and 20). All reported results with mean and standard deviation are computed over 3 independent runs.

Method	ImageNet-R (5 Tasks)		ImageNet-R (10 Tasks)		ImageNet-R (20 Tasks)	
	ACC_5	$\overline{ACC}_5$	ACC_{10}	$\overline{ACC}_{10}$	ACC_{20}	$\overline{ACC}_{20}$
L2P	$65.03_{\pm 0.03}$	$69.97_{\pm 0.15}$	$62.87_{\pm 0.72}$	$68.90_{\pm 0.58}$	$58.64_{\pm 0.34}$	$65.57_{\pm 0.35}$
DualPrompt	$68.24_{\pm 0.23}$	$71.82_{\pm 0.39}$	$65.30_{\pm 0.52}$	$69.62_{\pm 0.29}$	$60.47_{\pm 0.54}$	$65.91_{\pm 0.52}$
CODA-P	$73.65_{\pm 0.15}$	$77.88_{\pm 0.30}$	$72.10_{\pm 0.29}$	$76.90_{\pm 0.41}$	$67.16_{\pm 0.11}$	$72.34_{\pm 0.44}$
InfLoRA	$77.53_{\pm 0.30}$	$82.24_{\pm 0.11}$	$74.43_{\pm 0.31}$	$80.50_{\pm 0.06}$	$70.30_{\pm 0.14}$	$77.04_{\pm 0.06}$
SD-LoRA	$79.15_{\pm 0.20}$	$83.01_{\pm 0.42}$	$77.34_{\pm 0.35}$	$82.04_{\pm 0.24}$	$75.26_{\pm 0.37}$	$80.22_{\pm 0.72}$
VPT-NSP^2	$79.72_{\pm 0.19}$	$84.33_{\pm 0.29}$	$77.87_{\pm 0.10}$	$83.09_{\pm 0.26}$	$75.42_{\pm 0.27}$	$81.32_{\pm 0.21}$
CoSO	$\mathbf{82.10}_{\pm 0.13}$	$\mathbf{86.38}_{\pm 0.07}$	$\mathbf{81.10}_{\pm 0.39}$	$\mathbf{85.56}_{\pm 0.13}$	$\mathbf{78.19}_{\pm 0.28}$	$\mathbf{83.69}_{\pm 0.12}$



(a) 5 tasks (b) 10 tasks (c) 20 tasks

Figure 2: The detailed performance during the learning of ImageNet-R on (a) 5 tasks, (b) 10 tasks, and (c) 20 tasks.

Baselines and Implementation Details. We compare CoSO with several state-of-the-art PEFT-based methods: L2P [Wang et al., 2022b], DualPrompt [Wang et al., 2022a], CODA-Prompt (CODA-P) [Smith et al., 2023], InfLoRA [Liang and Li, 2024], VPT-NSP^2 [Lu et al., 2024], and SD-LoRA [Wu et al., 2025]. Comparing against both prompt-based and LoRA-based methods allows us to comprehensively evaluate the effectiveness of CoSO. In addition to the ViT-B/16 [Dosovitskiy et al., 2021] pretrained on ImageNet-1K, we also evaluate a self-supervised ViT-B/16 obtained with DINO [Caron et al., 2021]. Details of the experimental setup are provided in Appendix C.

4.2 Experimental Results

We evaluate CoSO against state-of-the-art continual learning methods across different experimental settings. Table 1 shows the performance comparison on ImageNet-R under various task partitions (5, 10, and 20 tasks). Across all partitions, CoSO delivers the highest final accuracy (ACC) and average accuracy ($\overline{ACC}$), confirming its robustness to mitigate forgetting. For the most challenging setting (20 tasks), CoSO attains 78.19% final accuracy and 83.69% average accuracy, while the best baseline method achieves 75.42% and 81.32%, respectively. For the ImageNet-R 10 tasks scenario, CoSO improves the final accuracy by 3.23% and the average accuracy by 2.47% compared to the best baseline method. Likewise, in the 5 tasks setting, CoSO still leads by 2.38% in final accuracy and 2.05% in average accuracy. This margin highlights CoSO's exceptional resistance to forgetting and its strong capacity to integrate new knowledge without eroding prior learning.

Figure 2 illustrates the evolution of accuracy throughout the continual learning process for various methods evaluated on ImageNet-R. It is evident that CoSO consistently maintains superior performance relative to other approaches, both during the intermediate phases and at the end of training. This ongoing superiority underscores CoSO's effectiveness in reducing interference from newly introduced tasks, resulting in a significantly slower decline in accuracy compared to competing methods. Complementary results in Table 2 reveal the same trend on CIFAR100 and DomainNet. On the DomainNet benchmark, CoSO outperforms the best baseline method by 1.75% in final accuracy and 1.37% in average accuracy, confirming its ability to generalize across heterogeneous visual domains.

Table 2: Results (%) on CIFAR100 (10 Tasks) and DomainNet (5 Tasks). All reported results with mean and standard deviation are computed over 3 independent runs.

Method	CIFAR100 (10 Tasks)		DomainNet (5 Tasks)	
	ACC_{10}	$\overline{ACC}_{10}$	ACC_5	$\overline{ACC}_5$
L2P	$82.64_{\pm 0.26}$	$87.90_{\pm 0.19}$	$70.03_{\pm 0.09}$	$75.65_{\pm 0.06}$
DualPrompt	$84.68_{\pm 0.22}$	$90.12_{\pm 0.05}$	$72.25_{\pm 0.05}$	$77.84_{\pm 0.02}$
CODA-P	$86.60_{\pm 0.37}$	$91.46_{\pm 0.20}$	$73.16_{\pm 0.07}$	$78.75_{\pm 0.04}$
InfLoRA	$86.85_{\pm 0.08}$	$91.45_{\pm 0.16}$	$73.09_{\pm 0.11}$	$79.21_{\pm 0.08}$
SD-LoRA	$87.30_{\pm 0.45}$	$91.81_{\pm 0.27}$	$73.20_{\pm 0.12}$	$79.03_{\pm 0.04}$
VPT-NSP^2	$88.09_{\pm 0.12}$	$92.48_{\pm 0.11}$	$72.52_{\pm 0.13}$	$78.68_{\pm 0.06}$
CoSO	$\mathbf{88.77}_{\pm 0.16}$	$\mathbf{92.99}_{\pm 0.23}$	$\mathbf{74.27}_{\pm 0.07}$	$\mathbf{80.05}_{\pm 0.04}$

Table 3: Ablation study results (%) on ImageNet-R with varying numbers of tasks (5, 10 and 20).

Method	ImageNet-R (5 Tasks)		ImageNet-R (10 Tasks)		ImageNet-R (20 Tasks)	
	ACC_5	$\overline{ACC}_5$	ACC_{10}	$\overline{ACC}_{10}$	ACC_{20}	$\overline{ACC}_{20}$
w/o Orth	79.35	85.22	75.90	83.43	69.75	78.88
w/o FD	80.72	85.44	78.83	84.45	76.68	82.41
CoSO	82.37	86.46	80.72	85.67	78.27	83.62

A detailed analysis of computational and memory costs are presented in Appendix D. The additional results with DINO [Caron et al., 2021] are provided in Appendix E.

Ablation Study. We conduct comprehensive ablation studies on ImageNet-R benchmark to validate the individual contributions of the orthogonal projection mechanism and the Frequent Directions (FD) based subspace consolidation. Specifically, we compare CoSO with two variants. The first variant (w/o Orth) removes the orthogonal projection, which directly uses the original gradients $G_{\tau,t}$ for optimization instead of the orthogonally projected gradients $G'_{\tau,t}$. This variant optimizes parameters in continuous subspaces without any orthogonality constraint, thereby ignoring task interference. The second variant (w/o FD) retains orthogonality but, instead of employing FD to consolidate all intermediate gradients from the current task, constructs the task-specific subspace using only the final subspace obtained at the end of that task.

The results are summarized in Table 3. Eliminating orthogonal projection (w/o Orth) leads to a sharp performance drop (8.52% in final accuracy) on 20 Tasks setting, highlighting the importance of excluding new gradients from the historical subspace to prevent interference. Replacing FD with the simplified strategy that builds each task-specific subspace from only the final gradient subspace (w/o FD) also degrades performance, lowering final accuracy by 1.65%, 1.89% and 1.59% for 5, 10 and 20 Tasks settings, respectively. This drop confirms that aggregating all intermediate gradients through incremental FD updates captures richer task information than using a single terminal subspace. Across the table, the full method delivers the highest final and average accuracies, indicating that both orthogonal projection and FD consolidation are indispensable for robust continual learning.

5 Conclusion

In this paper, we propose Continuous Subspace Optimization for Continual Learning (CoSO). CoSO optimizes the pre-trained models within continuous subspaces. By maintaining orthogonality between the current task's optimization subspace and that of historical tasks, CoSO effectively mitigates the interference. CoSO maintains a compact task-specific component while learning a task. After completing the current task, the task-specific component is used to update the historical task subspace. Extensive experiments on standard benchmarks demonstrate that CoSO consistently outperforms state-of-the-art baselines in both final accuracy and average accuracy over time, confirming its effectiveness and robustness across diverse data streams. In the future, a challenging open problem is to extend CoSO to multimodal task settings.

Acknowledgments and Disclosure of Funding

This work was partially supported by National Science and Technology Major Project (2022ZD0114801), and NSFC (U23A20382).

References

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision*, 2021.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2019a.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019b.

Romain Cosson, Ali Jadbabaie, Anuran Makur, Amirhossein Reisizadeh, and Devavrat Shah. Low-rank gradient descent. *IEEE Open Journal of Control Systems*, 2:380–395, 2023.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

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.

Yadin Dudai. The neurobiology of consolidations, or, how stable is the engram? *Annual review of psychology*, 55:51–86, 2004.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108, pages 3762–3773, 2020.

Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11483–11493, 2023.

Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8320–8329, 2021.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

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

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3925–3934, 2019.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

Yan-Shuo Liang and Wu-Jun Li. Adaptive plasticity improvement for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7816–7825, 2023.

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

Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30*, pages 6470–6479, 2017.

Yue Lu, Shizhou Zhang, De Cheng, Yinghui Xing, Nannan Wang, Peng Wang, and Yanning Zhang. Visual prompt tuning in null space for continual learning. In *Advances in Neural Information Processing Systems 37*, pages 7878–7901, 2024.

Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*, 2022.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. Unipelt: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

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

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

Quang Pham, Chenghao Liu, and Steven C. H. Hoi. Continual learning, fast and slow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):134–149, 2024.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5533–5542, 2017.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021.

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

Ghada Sokar, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Spacenet: Make free space for continual learning. *Neurocomputing*, 439:1–11, 2021.

Qing Sun, Fan Lyu, Fanhua Shang, Wei Feng, and Liang Wan. Exploring example influence in continual learning. In *Advances in Neural Information Processing Systems 35*, pages 27075–27086, 2022.

Gido M Van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010, 2017.

Yuanyu Wan and Lijun Zhang. Efficient adaptive online learning via frequent directions. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2748–2754, 2018.

Yuanyu Wan and Lijun Zhang. Approximate multiplication of sparse matrices with limited space. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 10058–10066, 2021.

Yuanyu Wan and Lijun Zhang. Efficient adaptive online learning via frequent directions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6910–6923, 2022.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(08):5362–5383, 2024.

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.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022b.

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

Wenhan Xia, Chengwei Qin, and Elad Hazan. Chain of lora: Efficient fine-tuning of language models via residual learning. *arXiv preprint arXiv:2401.04151*, 2024.

Sifan Yang, Yuanyu Wan, Peijia Li, Yibo Wang, Xiao Zhang, Zhewei Wei, and Lijun Zhang. Dimension-free adaptive subgradient methods with frequent directions. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 71249–71274, 2025.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3987–3995, 2017.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, pages 61121–61143, 2024.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*, pages 8363–8371, 2024a.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9851–9873, 2024b.

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: Please refer to Section 3 and 4.

   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: Please refer to Section 5.

   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: Please refer to Section 3 for the assumptions. Please refer to Appendixes for the complete proof.

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: Please refer to Section 4 and Appendix C.

   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 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: Please refer to the supplemental materials.

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: Please refer to Section 4 and Appendix C.

   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 main results with mean and standard deviation, which are computed over 3 independent runs.

   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: Please refer to Section 4 and Appendix C.

   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: The authors have read the NeurIPS Code of Ethics and ensured that our research conforms to it.

   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]

    Justification: This paper is about continual learning and does not involve societal impact.

    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: This paper is about continual learning and does not have a risk for misuse.

    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 have properly cited all data, code, and models used in this paper.

    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: [NA]

    Justification: The paper does not release new assets.

    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:

    - 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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: This research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Proof of Proposition 1

Recall that $\{G'_{\tau,t}\}_{t=1}^{T} \subset \mathbb{R}^{m\times n}$ is the sequence of projected gradients for task τ, $A_t = G'_{\tau,t}G'^{\top}_{\tau,t}$, $A = \sum_{t=1}^{T} G'_{\tau,t}G'^{\top}_{\tau,t}$, $\tilde{A}_t = Q_{\tau,t}Q^{\top}_{\tau,t}$, $\tilde{A} = \sum_{t=1}^{T} Q_{\tau,t}Q^{\top}_{\tau,t}$ and sketch matrix $S_{\tau,T} \in \mathbb{R}^{m\times r_2}$.

Because $Q_{\tau,t}$ is the rank r_2 approximation of $G'_{\tau,t}$, for every step t, we have

$$\|A_t - \tilde{A}_t\|_2 = \sigma_t^2, \tag{14}$$

where σ_t is the (r_2+1) singular value of $G'_{\tau,t}$.

Using the triangle inequality together with Eq. (14),

$$\|A - \tilde{A}\|_2 = \left\|\sum_{t=1}^{T}(A_t - \tilde{A}_t)\right\|_2 \leq \sum_{t=1}^{T}\|A_t - \tilde{A}_t\|_2 = \sum_{t=1}^{T}\sigma_t^2. \tag{15}$$

Since we use FD to compute $S_{\tau,T}$ based on $\{Q_{\tau,t}\}_{t=1}^{T}$, from Theorem 1.1 of Ghashami et al. [2016], we have

$$\|\tilde{A} - S_{\tau,T}S^{\top}_{\tau,T}\|_2 \leq \frac{\|\tilde{A} - [\tilde{A}]_k\|_F^2}{r_2 - k}, \tag{16}$$

where $[\tilde{A}]_k$ is the minimizer of $\|\tilde{A} - [\tilde{A}]_k\|_F$ overall rank k matrices. Applying the triangle inequality to $\|A - S_{\tau,T}S^{\top}_{\tau,T}\|$ and substituting Eq. (15) and (16) gives

$$\begin{aligned}\|A - S_{\tau,T}S^{\top}_{\tau,T}\|_2 &\leq \|A - \tilde{A}\|_2 + \|\tilde{A} - S_{\tau,T}S^{\top}_{\tau,T}\|_2 \\ &\leq \sum_{t=1}^{T}\sigma_t^2 + \frac{\|\tilde{A} - [\tilde{A}]_k\|_F^2}{r_2 - k},\end{aligned} \tag{17}$$

which is exactly (10).

B CoSO Algorithm

We present the the detailed procedure in Algorithm 1.

C Experimental Setups and Implementation Details

Following existing works [Smith et al., 2023, Wu et al., 2025], we adopt ViT-B/16 [Dosovitskiy et al., 2021] pre-trained on ImageNet-21K and fine-tuned on ImageNet-1K as our backbone model, which consists of 12 transformer blocks. For fair comparison, all methods use the same ViT-B/16 backbone and optimizer. Additionally, we also evaluate a self-supervised ViT-B/16 obtained with DINO [Caron et al., 2021]. The optimization is performed using Adam [Kingma, 2014] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The training epochs vary across datasets: 40 epochs for ImageNet-R, 20 epochs for CIFAR100, and 5 epochs for DomainNet. We maintain a consistent batch size of 128 across all experiments. Results are averaged over 3 independent runs, and we report the corresponding standard deviation. Notably, CoSO only optimize the output projection layers in multi-head attention module rather than QKV transformations.

We present the detailed hyperparameter settings of CoSO in Table 4. These hyperparameters are carefully tuned to balance memory efficiency and performance, reflecting the varying complexity of the datasets. The hyperparameter settings of baseline methods are following existing work [Wang et al., 2022a, Smith et al., 2023, Liang and Li, 2024, Lu et al., 2024, Wu et al., 2025]. For all datasets, we employ minimal data augmentation, consisting of random resized cropping to 224×224 pixels and random horizontal flipping during training, without any additional augmentation techniques. To prevent overfitting, we followed VPT-NSP^2 [Lu et al., 2024], setting the temperature parameter in the cross-entropy loss to 3 for all datasets. All experiments were conducted on NVIDIA A6000 GPUs with 48GB memory using PyTorch 2.5.1.

The projection rank (r_1) determines the dimensionality of the low-rank subspace for gradient projection. For simpler datasets like CIFAR100, a lower value of $r_1 = 15$ is sufficient, while more

Algorithm 1 CoSO for Continual Learning

Input: A layer weight matrix $W \in \mathbb{R}^{m\times n}$, step size η, decay rates β_1, β_2, projection rank r_1, FD rank r_2, threshold ϵ and update gap K.
Initialize first-order moment $M_0 \in \mathbb{R}^{m\times r} \leftarrow 0$
Initialize second-order moment $V_0 \in \mathbb{R}^{m\times r} \leftarrow 0$
Initialize sketch matrix $S_{\tau,0} \in \mathbb{R}^{m\times r} \leftarrow 0$
Initialize orthogonal projection matrix $\mathcal{M}_0 \leftarrow 0$
for Task $\tau \in 1 \dots N$ **do**
  for step $t \in 1 \dots T$ **do**
    $G_{\tau,t} \leftarrow \nabla_{W_{\tau,t}} L(W_{\tau,t})$ ▷ Compute mini-batch gradient for task τ
    $G'_{\tau,t} \leftarrow G_{\tau,t} - \mathcal{M}_{\tau-1}\mathcal{M}_{\tau-1}^\top G_{\tau,t}$ ▷ Orthogonal projection
    if $t \bmod K == 0$ **then**
      $U\Sigma V^\top = \text{SVD}(G'_{\tau,t})$
      $P_{\tau,t} = U[:, :r_1]$ ▷ Compute projection matrix $P_{\tau,t}$
      Update $S_{\tau,t}$ through Eq. (6) and (8) ▷ Use FD to consolidate gradient information
    else
      $P_{\tau,t} \leftarrow P_{\tau,t-1}$
      $S_{\tau,t} \leftarrow S_{\tau,t-1}$
    end if
    $R_{\tau,t} \leftarrow P_{\tau,t}^\top G'_{\tau,t}$ ▷ Project orthogonal gradient into low rank space
    Use $R_{\tau,t}$ to compute $N_{\tau,t}$ through Eq. (4) ▷ Update $R_{\tau,t}$ by Adam
    $\tilde{G}_{\tau,t} \leftarrow P_{\tau,t} N_{\tau,t}$ ▷ Project gradient back to original space
    $W_{\tau,t} \leftarrow W_{\tau,t-1} - \eta \cdot \tilde{G}_{\tau,t}$
  end for
  Update the historical subspaces basis matrix $\mathcal{M}_{\tau-1}$ through Eq. (11), (12) and (13)
end for

Table 4: Hyperparameter settings for different datasets.

Hyperparameter	CIFAR100	ImageNet-R	DomainNet
Projection rank (r_1)	15	50	70
Frequent directions rank (r_2)	100	120	160
Update gap (K)	1	1	20
Threshold (ϵ_{th})	0.98	0.98	0.98

complex datasets such as ImageNet-R and DomainNet require higher values (r_1 = 50 and r_1 = 70, respectively) to capture a richer set of gradient directions. The Frequent Directions rank (r_2) is consistently set higher than r_1 across all datasets. This design choice ensures that CoSO can capture a broader range of directions, reducing information loss during continual learning. As the dataset complexity increases, r_2 is adjusted upward to retain more task information.

The update gap K is adjusted based on the characteristics of each dataset. For DomainNet, we use a larger update gap ($K = 20$) due to its larger and more diverse task structure, where frequent updates may become redundant. In contrast, CIFAR100 and ImageNet-R exhibit rapid gradient changes, necessitating a smaller K. Finally, the threshold (ϵ) is uniformly set to 0.98 across all datasets. This value is selected to maintain a high retention rate of gradient information within the subspace.

D Analysis of Computational and Memory Costs

We conducted a comparative analysis of CoSO and baseline methods with respect to computational cost (reported as estimated GFLOPs) and memory usage, as summarized in Table 5. CoSO requires half the computational cost of prompt-based methods (such as L2P, DualPrompt, and CODA-P), as it avoids the need for twice forward passes through the network. In terms of memory usage, CoSO is on par with other low-rank adaptation techniques such as InfLoRA (13.44). Its slightly higher memory footprint (13.61) stems from using a larger rank for gradient subspace approximation, which enables better capture of task-specific patterns and leads to superior performance. Notably,

Table 5: Comparison on ImageNet-R (10 Tasks) in terms of computation (GFLOPs) and memory usage.

Method	GFLOPs	Memory Usage (G)
L2P	70.24	12.90
DualPrompt	70.24	12.96
CODA-P	70.24	12.97
InfLoRA	35.12	13.44
SD-LoRA	35.12	15.62
VPT-NSP^2	35.83	11.54
CoSO	35.12	13.61

Table 6: Results (%) on ImageNet-R (10 Tasks). All reported results with mean and standard deviation are computed over 3 independent runs.

Method (DINO)	ImageNet-R (10 Tasks)	
	ACC_{10}	$\overline{ACC}_{10}$
L2P	$61.94_{\pm 0.45}$	$68.77_{\pm 0.27}$
DualPrompt	$60.40_{\pm 0.18}$	$67.65_{\pm 0.07}$
CODA-P	$64.63_{\pm 0.33}$	$72.20_{\pm 0.30}$
InfLoRA	$67.91_{\pm 0.23}$	$76.40_{\pm 0.03}$
SD-LoRA	$69.78_{\pm 0.63}$	$65.73_{\pm 0.35}$
VPT-NSP^2	$69.68_{\pm 0.20}$	$77.24_{\pm 0.16}$
CoSO	$\mathbf{71.60}_{\pm 0.44}$	$\mathbf{79.28}_{\pm 0.16}$

simply increasing the rank for InfLoRA would not yield similar improvements, as its performance is limited by the constraint of fixed subspaces. Compared with SD-LoRA (15.62), which incurs the greatest memory overhead, CoSO offers a more efficient alternative while delivering competitive performance. Overall, these results highlight CoSO's ability to strike a favorable balance between computational efficiency and memory usage, making it a scalable solution for continual learning across diverse tasks.

E Additional Experiment Results on ImageNet-R

To further verify CoSO's generality, we test it on a self-supervised ViT-B/16 backbone trained with DINO [Caron et al., 2021] on ImageNet-R (10 Tasks). The results are presented in Table 6. CoSO outperforms the best baseline method with a considerable margin, confirming its ability to generalize across various vision transformers.