
Turning the Tables: Enabling Backward Transfer via Causal-Aware LoRA in Continual Learning

Chaoyang Li^{1,2}

Runze Ye¹

Jianyang Qin¹

Jinhao Cui¹

Lingzhi Wang¹

Ning Hu²

Qing Liao^{1,2*}

¹Harbin Institute of Technology, Shenzhen, China

²Peng Cheng Laboratory, Shenzhen, China

{22b951022, 24S151081, 22b351005, cuijinhao}@stu.hit.edu.cn,
{hun}@pcl.ac.cn, {wanglingzhi, liaqing}@hit.edu.cn

Abstract

Current parameter-efficient fine-tuning (PEFT) methods have shown superior performance in continual learning. However, most existing PEFT-based methods focus on mitigating catastrophic forgetting by limiting modifications to the old task model caused by new tasks. This hinders backward knowledge transfer, as when new tasks have a strong positive correlation with old tasks, appropriately training on new tasks can transfer beneficial knowledge to old tasks. Critically, achieving backward knowledge transfer faces two fundamental challenges: (1) some parameters may be ineffective on task performance, which constrains the task solution space and model capacity; (2) since old task data are inaccessible, modeling task correlation via shared data is infeasible. To address these challenges, we propose CaLoRA, a novel **causal-aware low-rank adaptation** framework that is the first PEFT-based continual learning work with backward knowledge transfer. Specifically, we first propose **parameter-level counterfactual attribution** (PaCA) that estimates the causal effect of LoRA parameters via counterfactual reasoning, identifying effective parameters from a causal view. Second, we propose **cross-task gradient adaptation** (CaGA) to quantify task correlation by gradient projection and evaluate task affinity based on gradient similarity. By incorporating causal effect, task correlation, and affinity, CaGA adaptively adjusts task gradients, facilitating backward knowledge transfer without relying on data replay. Extensive experiments across multiple benchmarks and continual learning settings show that CaLoRA outperforms state-of-the-art methods. In particular, CaLoRA better mitigates catastrophic forgetting by enabling positive backward knowledge transfer.

1 Introduction

Recently, parameter-efficient fine-tuning (PEFT) [1–7] has emerged as a promising technology for continual learning (CL), offering a feasible solution to maintain performance on old tasks and improving learning capabilities for new ones [8]. Among the most widely adopted techniques include prompt-tuning [1–4], adapter-tuning [9–11], and low-rank adaptation (LoRA) [6, 7, 12, 13], all of which primarily address catastrophic forgetting [1, 6, 9, 12]. These methods employ two fundamental

*Corresponding author.

Table 1: Comparison of PEFT-based continual learning methods.

Method	Data Replay-Free	Efficient Inference	Backward Transfer
CodaPrompt [20](CVPR’23)	✓	×	×
HidePrompt [21](NeurIPS’23)	×	×	×
InfLoRA [6] (CVPR’24)	✓	✓	×
SAPT [13] (ACL’24)	×	✓	×
SD-LoRA [7] (ICLR’25)	✓	✓	×
CaLoRA(ours)	✓	✓	✓

strategies: selectively incorporating old task parameters during new task learning [1, 3, 7, 9, 13], or ensuring orthogonal updates to new task parameters relative to old ones [6, 12].

Although most existing continual learning methods based on PEFT effectively mitigate catastrophic forgetting [1, 3, 6, 9], the parameter-freezing strategy inherently limits backward knowledge transfer by treating updates for new tasks as potential interference with old tasks [6, 12, 14]. The learning parameters of new tasks do not necessarily conflict with those of old tasks [15, 16]. When the new task is positively correlated with old ones, it can facilitate backward knowledge transfer by contributing beneficial information to old tasks [16]. However, existing continual learning methods based on PEFT have not yet explored the backward knowledge transfer, as shown in the Table. 1.

This motivates us to propose a novel continual learning approach based on PEFT that addresses forgetting and enhances old tasks through effective backward knowledge transfer. To achieve this, it is essential to understand when and how backward knowledge transfer occurs. When a new task is strongly correlated with old tasks, it creates favorable conditions for knowledge transfer [15, 16]. In this case, if this transfer is positive (i.e., the new task can transfer beneficial knowledge to old tasks), we can promote backward knowledge transfer by optimizing parameters beneficial to old tasks during the training of the new task. Conversely, if the transfer is negative (i.e., the new task interferes with old tasks), it is necessary to protect old task knowledge to mitigate catastrophic forgetting.

However, realizing the above objective faces two key challenges:(1) Although existing PEFT methods can achieve performance comparable to full-parameter training by tuning only a few parameters, some parameters may be ineffective in enhancing performance [17–19]. Moreover, as the number of tasks increases, the parameter solution space for new tasks becomes increasingly constrained by old tasks, leading to worse performance [15]. (2) Since data from old tasks is not replayable, traditional methods for modeling task correlation via shared data are infeasible [7, 15, 16].

To address these challenges, we propose CaLoRA, a novel **causal-aware low-rank adaptation** framework, which is the first PEFT-based continual learning work to enable backward knowledge transfer. Specifically, to tackle the first challenge, we propose the **parameter-level counterfactual attribution** (PaCA) method that estimates the causal effect of LoRA parameters via counterfactual reasoning. PaCA enables the model to identify effective parameters with strong causal effects for the current task while reducing the influence of ineffective ones, enhancing its ability to learn future tasks. To address the second challenge, we propose the **cross-task gradient adaptation** (CaGA) method, which measures task correlation by gradient projection and computes task affinity based on gradient similarity in the task space. By jointly leveraging causal effect, task correlation, and affinity, CaGA adaptively adjusts task gradients, enabling effective backward knowledge transfer without data replay. The key contributions of this paper are summarized as follows:

- We propose CaLoRA, a novel **causal-aware low-rank adaptation** framework, which is the first PEFT-based continual learning work to mitigate catastrophic forgetting more effectively by actively enabling backward knowledge transfer.
- We propose parameter-level counterfactual attribution to identify effective parameters via counterfactual reasoning, reducing the influence of ineffective parameters in current tasks and expanding the capacity for future tasks.
- We propose cross-task gradient adaptation that adaptively adjusts task gradients based on causal effect, task correlation, and affinity, enabling backward knowledge transfer without data replay.
- Extensive experiments on multiple benchmarks under various task settings demonstrate that CaLoRA consistently outperforms existing state-of-the-art methods in continual learning.

2 Related Work

2.1 Parameter-Efficient Fine-Tuning.

Parameter-efficient fine-tuning (PEFT) methods involve freezing a pre-trained model and introducing fewer parameters than full fine-tuning, often achieving comparable or even superior performance [22, 23]. For instance, the Adapter [9–11, 18, 24] incorporates small modules at various layers of the pretrained model, tuning only these additional modules for task adaptation. Prompt-tuning [4, 25] and prefix-tuning [26] introduce learnable tokens into the input for the pretrained model, adjusting only these tokens to fit specific tasks. Low-rank adaptation (LoRA) [19, 22, 27] reparameterizes pre-trained weights with two low-rank matrices, tuning solely these matrices for downstream adaptation.

2.2 Continual Learning.

Continual learning focuses on sequentially acquiring knowledge from new tasks while maintaining performance on old ones [28, 29]. Existing approaches primarily fall into three categories: replay-based, regularization-based, and expansion-based methods. Replay-based methods [30–34] mitigate forgetting by storing and reusing samples from old tasks. Regularization-based techniques [35–37] constrain updates to parameters critical for old tasks through designing a penalty term. Expansion-based approaches [38, 39] dynamically adjust model capacity to accommodate new tasks while preserving learned knowledge.

Recently, PEFT has shown effectiveness in continual learning by mitigating catastrophic forgetting, including prompt-tuning, adapter-tuning, and LoRA. Specifically, prompt-tuning methods [1–3] introduce learnable prompt tokens into transformer layers. For example, L2P [1], DualPrompt [40], and CODA-Prompt [20] integrate Vision Transformers (ViTs) with learnable embeddings with prompts, thereby improving knowledge retention as new tasks are learned. Building on these, HiDe-Prompt [21] further stores old task samples to enhance performance. Adapter-tuning methods [10, 11] often insert lightweight learnable modules into transformer layers, like C-ADA [11] designs a parameter-extensible continual adapter layer in the pre-trained model. LoRA [12, 22] updates the pre-trained weights with low-rank matrices. For instance, InfLoRA [6] prevents interference between tasks by constraining new task updates to the orthogonal space of old tasks. SD-LoRA [7] decouples the learning of the magnitude and direction of LoRA components. Moreover, SAPT [13] aligns the PEFT block (including prompt and LoRA) via a shared attentive module and replays old task samples to improve performance. Despite these advances, existing PEFT-based continual learning methods fail to simultaneously satisfy all desirable properties outlined in Table 1. To bridge this gap, we propose CaLoRA, a novel **causal-aware low-rank adaptation** framework that enables backward knowledge transfer without data replay.

3 Methodology

Fig. 1 illustrates the CaLoRA framework, which incorporates **parameter-level counterfactual attribution** (PaCA) and **cross-task gradient adaptation** (CaGA). First, PaCA estimates causal effects of LoRA parameters via counterfactual reasoning, which identifies effective parameters from a causal view. Second, CaGA measures task correlation and affinity to determine conditions for knowledge transfer. CaGA further adapts gradients based on causal effects, task correlation, and affinity, thereby promoting beneficial knowledge transfer. More details are in subsections 3.2 and 3.3.

3.1 Preliminaries

Problem Definition. Continual learning tackles the challenge of sequentially learning from multiple tasks, where each task $\mathcal{T}_t$ in the sequence $\{\mathcal{T}_1, \dots, \mathcal{T}_T\}$ is associated with a distinct dataset $\mathcal{D}_t = \{(x_t^j, y_t^j)\}_{j=1}^{|\mathcal{D}_t|}$. Here, x_t^j denotes an input sample and y_t^j its corresponding label, with $|\mathcal{D}_t|$ representing the dataset size. The model f_Θ , parameterized by Θ , is trained exclusively on $\mathcal{D}_t$ when learning task $\mathcal{T}_t$. The training objective for the t -th task is defined as (1):

$$\mathcal{L}(\mathcal{D}_t; \Theta) = \frac{1}{|\mathcal{D}_t|} \sum_{j=1}^{|\mathcal{D}_t|} \mathcal{L}(f_\Theta(x_t^j), y_t^j), \quad (1)$$

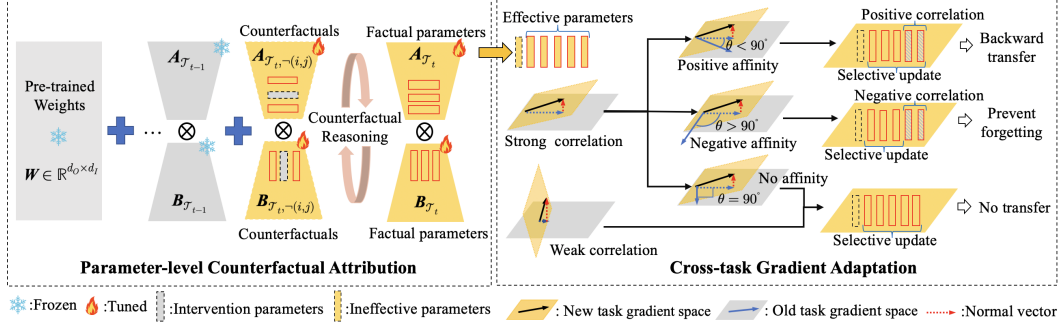


Figure 1: The CaLoRA framework consists of two components. First, Parameter-level Counterfactual Attribution estimates causal effects of parameters via counterfactual reasoning, identifying effective parameters from a causal view. Second, Cross-task Gradient Adaptation assesses task correlation and affinity to determine conditions for effective knowledge transfer. It further adapts gradients based on causal effects, task correlation, and affinity, thereby promoting backward knowledge transfer.

where $\mathcal{L}$ denotes the loss function. The ultimate goal in continual learning is to train the model f_{Θ} to ensure good performance not only on the new task $\mathcal{T}_t$ but also on all old tasks $\{\mathcal{T}_1, \dots, \mathcal{T}_{t-1}\}$.

Revisit Low-Rank Adaptation. LoRA [22] aims to constrain parameter updates to a low-rank subspace during fine-tuning. Given a pre-trained weight matrix $\mathbf{W} \in \mathbb{R}^{d_O \times d_I}$ for a specific pretrained layer with input dimension d_I and output dimension d_O , LoRA re-parameterizes it by introducing two low-rank matrices: $\mathbf{A} \in \mathbb{R}^{r \times d_I}$ and $\mathbf{B} \in \mathbb{R}^{d_O \times r}$, where $r \ll \min(d_I, d_O)$. Here, $\mathbf{A}$ reduces the dimensionality, and $\mathbf{B}$ restores it. The modified forward pass becomes $\mathbf{z} = \mathbf{W}\mathbf{h} + \mathbf{B}\mathbf{A}\mathbf{h}$, where $\mathbf{h}$ is the input and $\mathbf{z}$ is the output. LoRA initializes $\mathbf{B}$ to zeros and $\mathbf{A}$ with a Gaussian distribution, keeping $\mathbf{W}$ fixed while fine-tuning only $\mathbf{A}$ and $\mathbf{B}$. To achieve the goal of continual learning, given the task sequence $\{\mathcal{T}_1, \dots, \mathcal{T}_T\}$, each task has two trainable matrices. The output of the t -th task at the specific layer is modified as Eq. (2). The LoRA of the t -th task updates parameter matrices $\mathbf{A}_{\mathcal{T}_t}$ and $\mathbf{B}_{\mathcal{T}_t}$ with gradient (i.e., $\mathbf{G}_{\mathcal{T}_t} = \nabla_{\mathbf{W}_{\mathcal{T}_t}} \mathcal{L}(\mathcal{D}_{\mathcal{T}_t}; \mathbf{W}_{\mathcal{T}_t})$) for every step s by Eq.(3).

$$\mathbf{z}_{\mathcal{T}_t} = \mathbf{W}_{\mathcal{T}_t} \mathbf{h} = \mathbf{W} \mathbf{h} + \sum_{k=1}^t \mathbf{B}_{\mathcal{T}_k} \mathbf{A}_{\mathcal{T}_k} \mathbf{h} = \mathbf{W}_{\mathcal{T}_{t-1}} \mathbf{h} + \mathbf{B}_{\mathcal{T}_t} \mathbf{A}_{\mathcal{T}_t} \mathbf{h}, \quad (2)$$

$$\mathbf{A}_{\mathcal{T}_t, (s+1)} \leftarrow \mathbf{A}_{\mathcal{T}_t, (s)} - \eta \mathbf{B}_{\mathcal{T}_t, (s)}^\top \mathbf{G}_{\mathcal{T}_t}, \quad \mathbf{B}_{\mathcal{T}_t, (s+1)} \leftarrow \mathbf{B}_{\mathcal{T}_t, (s)} - \eta \mathbf{G}_{\mathcal{T}_t} \mathbf{A}_{\mathcal{T}_t, (s)}^\top. \quad (3)$$

3.2 Parameter-level Counterfactual Attribution

Although LoRA constrains parameter updates to a low-rank subspace, existing studies have demonstrated that such low-rank constraints do not guarantee that all parameters are effective in improving task performance [17–19]. In continual learning scenarios, this limitation becomes particularly challenging because new task parameters are typically constrained (e.g., via gradient orthogonality [6, 12, 14]) to protect old tasks. It is worth noting that when ineffective parameters from old tasks are used to constrain new tasks, it not only fails to alleviate catastrophic forgetting but may also harm new task performance [15]. To address this, we propose a **parameter-level counterfactual attribution** (PaCA) method that uses counterfactual reasoning to assess the causal effects of low-rank parameters, identifying the effective parameters with strong causal effect.

Specifically, to quantify the causal effect of an individual parameter $\mathbf{W}_{\mathcal{T}_t, (i,j)}$ in the parameter matrix $\mathbf{W}_{\mathcal{T}_t} \in \mathbb{R}^{d_O \times d_I}$, we first perform a causal intervention by setting the parameter to zero [41–43], generating the counterfactual weight matrix $\mathbf{W}_{\mathcal{T}_t, \neg(i,j)}$. The individual causal effect is then defined as the change in loss when a parameter is included versus excluded [44–46], as shown in Eq. (4):

$$\mathbf{E}_{\mathcal{T}_t, (i,j)} = \mathcal{L}(\mathcal{D}_t; do(\mathbf{W}_{\mathcal{T}_t, (i,j)} = 0)) - \mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t}) = \mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t, \neg(i,j)}) - \mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t}), \quad (4)$$

where $do(\mathbf{W}_{\mathcal{T}_t, (i,j)} = 0)$ refers to a causal intervention operation that sets $\mathbf{W}_{\mathcal{T}_t, (i,j)}$ to zero. However, directly computing the causal effect by intervening on each parameter is computationally prohibitive. To address this, we approximate the counterfactual loss $\mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t, \neg(i,j)})$ using a second-order Taylor

expansion [47] around the original weight matrix $\mathbf{W}_{\mathcal{T}_t}$. This leads to the quadratic approximation shown in Eq. (5):

$$\begin{aligned} \mathbf{E}_{\mathcal{T}_t, (i,j)} &\approx \mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t}) - \frac{\partial \mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t})}{\partial \mathbf{W}_{\mathcal{T}_t, (i,j)}} \Delta \mathbf{W}_{\mathcal{T}_t, (i,j)} + \frac{1}{2} \frac{\partial^2 \mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t})}{\partial \mathbf{W}_{\mathcal{T}_t, (i,j)}^2} \Delta \mathbf{W}_{\mathcal{T}_t, (i,j)}^2 - \mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t}) \\ &= - \frac{\partial \mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t})}{\partial \mathbf{W}_{\mathcal{T}_t, (i,j)}} \Delta \mathbf{W}_{\mathcal{T}_t, (i,j)} + \frac{1}{2} \frac{\partial^2 \mathcal{L}(\mathcal{D}_t; \mathbf{W}_{\mathcal{T}_t})}{\partial \mathbf{W}_{\mathcal{T}_t, (i,j)}^2} \Delta \mathbf{W}_{\mathcal{T}_t, (i,j)}^2, \end{aligned} \quad (5)$$

where $\mathbf{E}_{\mathcal{T}_t} \in \mathbb{R}^{d_O \times d_I}$, $\Delta \mathbf{W}_{\mathcal{T}_t, (i,j)} = \mathbf{W}_{\mathcal{T}_t, (i,j)} - \mathbf{W}_{\mathcal{T}_t, \neg(i,j)}$. We define the first-order causal effect as the term involving only the first-order derivative, while the second-order causal effect includes both first- and second-order terms. Since directly computing the second-order term incurs quadratic complexity, we employ a diagonal Hessian approximation in practice [48], which reduces the computational cost to linear time [47]. This way enables scalable computation for both causal effect approximations. When $\mathbf{E}_{\mathcal{T}_t, (i,j)}$ is positive, it indicates that removing the parameter will lead to an increase in the task loss, suggesting that the parameter is causally essential for improving task performance. Conversely, when $\mathbf{E}_{\mathcal{T}_t, (i,j)}$ is negative, the parameter is ineffective. Finally, to map the causal effect $\mathbf{E}_{\mathcal{T}_t}$ to a weight between 0 and 1, we apply the softmax function to get $\hat{\mathbf{E}}_{\mathcal{T}_t} = \text{Softmax}(\mathbf{E}_{\mathcal{T}_t})$. A larger value of $\hat{\mathbf{E}}_{\mathcal{T}_t}(i, j)$ reflects a stronger causal effect of the parameter on task performance. Prioritizing such effective parameters during the current task training not only reduces the influence of ineffective ones but also enhances the model’s ability to learn further tasks.

3.3 Cross-task Gradient Adaptation

After obtaining the causal effects of the parameters, another key challenge is accurately estimating the correlation between the new task and old tasks. This is crucial to ensure that the causal effects can be effectively integrated with task correlation to refine old tasks appropriately. To achieve this, we propose a **cross-task gradient adaptation** (CaGA) method that selectively updates parameters beneficial to old tasks when learning a new one. Specifically, we first quantify task correlation through cross-task gradient projection to evaluate the possibility of knowledge transfer. Then, we compute task affinity via gradient similarity to assess whether the transfer is beneficial (positive transfer) or detrimental (negative transfer). Based on these measures, we identify the gradient components of the new task that are most advantageous to old tasks. Finally, we perform a targeted update by emphasizing the gradient components that are causally effective for the new task and those that are beneficial to old tasks, thereby enabling potential backward knowledge transfer.

Task Correlation. To determine whether the new task meets the conditions for backward knowledge transfer to old tasks, we characterize the correlation between the input subspaces of old and new tasks using gradient projection [15, 16]. Specifically, given any new task $\mathcal{T}_t$ ($t \in [2, T]$) with gradient $\mathbf{G}_{\mathcal{T}_t} \in \mathbb{R}^{d_O \times d_I}$, its correlation with a old task $\mathcal{T}_k$ ($k \in [1, t-1]$) is defined as $C_{\mathcal{T}_t \rightarrow \mathcal{T}_k}$:

$$\mathbf{P}_{\mathcal{T}_t \rightarrow \mathcal{T}_k} = \mathbf{U}_{\mathcal{T}_k} \mathbf{U}_{\mathcal{T}_k}^\top \mathbf{G}_{\mathcal{T}_t}, \quad C_{\mathcal{T}_t \rightarrow \mathcal{T}_k} = \frac{\|\mathbf{P}_{\mathcal{T}_t \rightarrow \mathcal{T}_k}\|_2}{\|\mathbf{G}_{\mathcal{T}_t}\|_2}, \quad C_{\mathcal{T}_t \rightarrow \mathcal{T}_k} \in (0, 1), \quad (6)$$

where $\mathbf{P}_{\mathcal{T}_t \rightarrow \mathcal{T}_k} \in \mathbb{R}^{d_O \times d_I}$ denotes the projection of the gradient of $\mathcal{T}_t$ onto the input subspace of $\mathcal{T}_k$, and $\mathbf{U}_{\mathcal{T}_k} = [\mathbf{u}_1, \dots, \mathbf{u}_r] \in \mathbb{R}^{d_O \times r}$ is the base for the input subspace of $\mathcal{T}_k$ obtained via singular value decomposition (SVD) (see Appendix A for details). Intuitively, the norm of projected gradient (i.e., $\|\mathbf{P}_{\mathcal{T}_t \rightarrow \mathcal{T}_k}\|_2$) serves as a proxy for the correlation between the input subspaces of the two tasks, because the gradient lies within the span of the input features [6, 15, 16, 49]. A higher correlation score $C_{\mathcal{T}_t \rightarrow \mathcal{T}_k}$ indicates that $\mathbf{G}_{\mathcal{T}_t}$ projects strongly onto the input subspace of $\mathcal{T}_k$, suggesting substantial input subspace overlap and favorable conditions for backward knowledge transfer [15, 16].

Task Affinity. We introduce task affinity to assess whether the knowledge transfer from the new task to old tasks is beneficial. Since the similarity of task gradients within a shared parameter space can reflect inter-task relationships [46, 50], we quantify these relationships by computing the directional similarity between the projected gradient component of the new task onto the input space of the old tasks and the gradient of the old tasks themselves. Specifically, for any new task $\mathcal{T}_t$ ($t \in [2, T]$), we define the task affinity matrix $\hat{\mathbf{S}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k} \in \mathbb{R}^{d_O \times d_I}$ to denote whether it has a positive transfer to the

old task $\mathcal{T}_k (k \in [1, t-1])$. The j -th column vector $\hat{\mathbf{S}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)}$ is defined as Eq. (7):

$$\mathbf{S}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)} = \frac{\mathbf{P}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)}}{\|\mathbf{P}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)}\|_2} \cdot \frac{\mathbf{G}_{\mathcal{T}_k, (:, j)}}{\|\mathbf{G}_{\mathcal{T}_k, (:, j)}\|_2}, \quad \hat{\mathbf{S}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)} = \begin{cases} +\mathbf{1}_{d_O} \in \mathbb{R}^{d_O}, & \text{if } \mathbf{S}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)} > 0 \\ \mathbf{0}_{d_O} \in \mathbb{R}^{d_O}, & \text{if } \mathbf{S}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)} = 0 \\ -\mathbf{1}_{d_O} \in \mathbb{R}^{d_O}, & \text{if } \mathbf{S}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)} < 0 \end{cases} \quad (7)$$

where $\mathbf{G}_{\mathcal{T}_k, (:, j)}$ denotes the j -th column vector of the gradient of $\mathcal{T}_k$ and $\mathbf{S}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)} \in [-1, 1]$ measures the similarity between the j -th column vectors of $\mathbf{P}_{\mathcal{T}_t \rightarrow \mathcal{T}_k}$ and $\mathbf{G}_{\mathcal{T}_k}$ in the input subspace of $\mathcal{T}_k$. The set $\{+\mathbf{1}_{d_O}, \mathbf{0}_{d_O}, -\mathbf{1}_{d_O}\}$ denotes all possible values of task affinity, where each is a d_O -dimensional column vectors with entries in $\{+1, 0, -1\}$, used to indicate the affinity between corresponding columns of the new and old task gradients.

Adaptive Gradient Correction. To enable effective backward knowledge transfer, we propose an adaptive gradient correction strategy to selectively update parameters under three cases: (1) If the new task exhibits no significant correlation to old tasks (indicating no potential for backward knowledge transfer) or the task affinity is zero, the strategy focuses on updating effective parameters with strong causal effects, without considering the impact on old tasks; (2) If the new task is strongly correlated to the old tasks (indicating potential for transfer), but the task affinity is negative, suggesting that the new task interferes with the old tasks, the strategy protects the parameters of the old tasks to prevent catastrophic forgetting; (3) If the new task demonstrates both strong correlation and positive task affinity with the old tasks (enabling effective backward knowledge transfer), it selectively updates the parameters that benefit the old tasks, thereby facilitating knowledge migration. To implement these adaptive updates, we introduce a gradient correction term, denoted as $\hat{\mathbf{G}}_{\mathcal{T}_t}$:

$$\hat{\mathbf{P}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k} = C_{\mathcal{T}_t \rightarrow \mathcal{T}_k} \cdot \frac{\mathbf{P}_{\mathcal{T}_t \rightarrow \mathcal{T}_k}}{\|\mathbf{P}_{\mathcal{T}_t \rightarrow \mathcal{T}_k}\|_2}, \quad \hat{\mathbf{G}}_{\mathcal{T}_t} = \hat{\mathbf{E}}_{\mathcal{T}_t} \circ \mathbf{G}_{\mathcal{T}_t} \circ \left[\mathbf{1}_{d_O \times d_I} + \sum_{k=1}^{t-1} \hat{\mathbf{S}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k} \circ \hat{\mathbf{P}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k} \right], \quad (8)$$

where $\hat{\mathbf{P}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k}$ denotes the gradient projection of $\mathcal{T}_t$, weighted by its correlation with the input space of $\mathcal{T}_k$, capturing the extent to which optimizing the parameter space of $\mathcal{T}_t$ affects that of $\mathcal{T}_k$. $\mathbf{1}_{d_O \times d_I}$ denotes an all-ones matrix of dimension $d_O \times d_I$, $\cdot$ indicates scalar multiplication, and $\circ$ represents the element-wise (Hadamard) product of matrices. Finally, the LoRA of task $\mathcal{T}_t$ updates its parameter matrices $\mathbf{A}_{\mathcal{T}_t}$ and $\mathbf{B}_{\mathcal{T}_t}$ at each training step s using the corrected gradient defined in Eq. (9).

$$\mathbf{A}_{\mathcal{T}_t, (s+1)} \leftarrow \mathbf{A}_{\mathcal{T}_t, (s)} - \eta \mathbf{B}_{\mathcal{T}_t, (s)}^\top \hat{\mathbf{G}}_{\mathcal{T}_t}, \quad \mathbf{B}_{\mathcal{T}_t, (s+1)} \leftarrow \mathbf{B}_{\mathcal{T}_t, (s)} - \eta \hat{\mathbf{G}}_{\mathcal{T}_t} \mathbf{A}_{\mathcal{T}_t, (s)}^\top. \quad (9)$$

In the Eq. (8), the corrected gradient $\hat{\mathbf{G}}_{\mathcal{T}_t}$ can adaptively adjust the gradient of task $\mathcal{T}_t$ at each iteration to achieve better knowledge transfer. Let $\mathbf{G}_{\mathcal{T}_t, (:, j)}$ denote the j -th column vector of the gradient for task $\mathcal{T}_t$. When the values of $\hat{\mathbf{P}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)}$ are very small (i.e., $\hat{\mathbf{G}}_{\mathcal{T}_t, (:, j)} \approx \hat{\mathbf{E}}_{\mathcal{T}_t, (:, j)} \circ \mathbf{G}_{\mathcal{T}_t, (:, j)}$) or when $\hat{\mathbf{S}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)}$ is a zero vector (i.e., $\hat{\mathbf{G}}_{\mathcal{T}_t, (:, j)} = \hat{\mathbf{E}}_{\mathcal{T}_t, (:, j)} \circ \mathbf{G}_{\mathcal{T}_t, (:, j)}$), it indicates that the new task has no significant influence on old tasks. In this case, parameter updates based on $\hat{\mathbf{G}}_{\mathcal{T}_t, (:, j)}$ can proceed only based on causal effects, without interfering with old task knowledge. In contrast, when $\hat{\mathbf{P}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)}$ contains large values and $\hat{\mathbf{S}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)}$ is a vector of -1, it indicates that a conflict between new and old tasks. To resolve this, the corrected gradient is computed as $\hat{\mathbf{G}}_{\mathcal{T}_t, (:, j)} = \hat{\mathbf{E}}_{\mathcal{T}_t, (:, j)} \circ \left[\mathbf{G}_{\mathcal{T}_t, (:, j)} - \sum_{k=1}^{t-1} \hat{\mathbf{P}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)} \circ \mathbf{G}_{\mathcal{T}_t, (:, j)} \right]$, which removes gradient components aligned with old tasks. Updating the parameters based on this residual gradient helps mitigate catastrophic forgetting. Finally, when the values in $\hat{\mathbf{P}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)}$ are large, and $\hat{\mathbf{S}}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (:, j)}$ is a vector of 1, it indicates that the new task positively contributes to old tasks. In this case, $\hat{\mathbf{G}}_{\mathcal{T}_t, (:, j)}$ selectively incorporates advantageous components from $\mathbf{G}_{\mathcal{T}_t, (:, j)}$ to promote backward knowledge transfer.

3.4 Algorithm and Time Complexity

Algorithm. CaLoRA adaptively updates the LoRA parameters based on the gradient correction terms during the training process. As shown in Algorithm 1, given the data $\mathcal{D}_t$ of the current task t , CaLoRA first estimates the causal effect of the LoRA parameters at the current step s . For new tasks $\mathcal{T}_t (t > 1)$, it calculates the correlation and affinity between the new task and the old tasks at the

Algorithm 1 CaLoRA for Continual Learning

Input: datasets: $\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^T$, pretrained model $f_{\mathbf{W}}(\cdot)$, training steps S
Output: model $f_{\mathbf{W}}(\cdot)$ with learned parameters $\{\mathbf{W}_{\mathcal{T}_t}\}_{t=1}^T$

- 1: **for** $\mathcal{D}_t$ in $\mathcal{D}$ **do**
- 2: Initialize $\mathbf{A}_{\mathcal{T}_t}$ and $\mathbf{B}_{\mathcal{T}_t}$.
- 3: **for** $s \in [1, \dots, S]$ **do**
- 4: Estimate causal effect $\hat{\mathbf{E}}_{\mathcal{T}_t, (s)}$ at step s by $\hat{\mathbf{E}}_{\mathcal{T}_t, (s)} = \text{Softmax}(\mathbf{E}_{\mathcal{T}_t, (s)})$.
- 5: **if** $t > 1$ **then**
- 6: Measure task correlation $C_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (s)}$ at step s by Eq. (6).
- 7: Quantify task affinity $\hat{S}_{\mathcal{T}_t \rightarrow \mathcal{T}_k, (s)}$ at step s by Eq. (7).
- 8: Calculate gradient correction term $\hat{\mathbf{G}}_{\mathcal{T}_t, (s)}$ at step s by Eq. (8).
- 9: **end if**
- 10: Update $\mathbf{A}_{\mathcal{T}_t, (s+1)}$, $\mathbf{B}_{\mathcal{T}_t, (s+1)}$ using Eq. (9).
- 11: Save the optimal task gradient in the task memory.
- 12: **end for**
- 13: **end for**

current step s . Finally, CaLoRA adaptively updates the LoRA parameters based on the causal effect, task correlation, and affinity.

Time Complexity Analysis. Given a pretrained weight matrix $\mathbf{W} \in \mathbb{R}^{d_O \times d_I}$, PaCA estimates causal effects via a second-order Taylor expansion with a diagonal Hessian approximation. Both the first-order and second-order terms have a time complexity of $\mathcal{O}(d_O \times d_I)$, making the overall complexity of PaCA linear and comparable to a standard backward pass. This computation can be efficiently parallelized and seamlessly integrated into the training process. In a task sequence of T tasks, for each new task, each gradient projection and affinity computation costs $\mathcal{O}(d_O \times d_I)$, leading to a total overhead of $\mathcal{O}((t-1) \times d_O \times d_I)$. Thus, the per-task complexity is approximately $\mathcal{O}(t \times d_O \times d_I)$. The total complexity for the entire sequence is $\mathcal{O}\left(\sum_{t=1}^T t \times d_O \times d_I\right)$.

4 Experiment

4.1 Experimental Setups

Dataset. We conduct experiments on two natural language processing (NLP) benchmarks (i.e., SuperNI [51] and Long Sequence [52]) and a computer vision (CV) benchmark (i.e., ImageNet-R [53]). Specifically, following prior work [13], both the SuperNI and Long Sequence benchmarks consist of 15 tasks, each evaluated under two different task orders. ImageNet-R is derived from 200 ImageNet classes through various artistic transformations. For the ImageNet-R benchmark, in line with previous studies [6, 7, 20], it is split into 5, 10, and 20 tasks, corresponding to 40, 20, and 10 classes per task, respectively. Further details on the benchmarks are provided in Appendix B.

Metrics. Let $\text{Acc}_{\mathcal{T}_t, \mathcal{T}_k}$ denote the testing performance (Accuracy for classification tasks and Rouge-L [54] for other tasks, more details are provided in the Appendix B) on the k -th task after training t -th task. The evaluation metrics are defined as follows: (1) Average Performance (AP) [55] reflects the mean performance across all tasks after training on the final task, i.e., $AP = \frac{1}{T} \sum_{t=1}^T \text{Acc}_{\mathcal{T}_T, \mathcal{T}_t}$; (2) Forgetting Rate (FRa) [55] denotes the extent of knowledge forgotten across the first $t-1$ tasks, i.e., $FRa = \frac{1}{T-1} \sum_{t=1}^{T-1} (\max_{k=1}^{T-1} \text{Acc}_{\mathcal{T}_k, \mathcal{T}_t} - \text{Acc}_{\mathcal{T}_T, \mathcal{T}_t})$; (3) Forward Transfer (FWT) [56] measures the influence of old task knowledge on learning new tasks, i.e., $FWT = \frac{1}{T} \sum_{t=1}^T \text{Acc}_{\mathcal{T}_t, \mathcal{T}_t} - \text{Acc}_{\mathcal{T}_t}$, where $\text{Acc}_{\mathcal{T}_t}$ refers to the performance of training task $\mathcal{T}_t$ individually; (4) Backward Transfer (BWT) [57] assesses the impact of learning new tasks on old tasks, i.e., $BWT = \frac{1}{T-1} \sum_{t=1}^{T-1} (\text{Acc}_{\mathcal{T}_T, \mathcal{T}_t} - \text{Acc}_{\mathcal{T}_t, \mathcal{T}_t})$.

Comparison Baselines and Training Details. We evaluate CaLoRA against nine PEFT-based continual learning baselines, including SeqLoRA, L2P [1], CodaPrompt [20], HidePrompt [21], O-LoRA [12], InfLoRA [6], SAPT-P [13], SAPT-LoRA [13], and SD-LoRA [7]. CaLoRA is a

Table 2: Overall results (mean $\pm$ std over 3 random seeds) on SuperNI benchmark with two task orders, evaluated using the T5-Large model. **Bold** indicates the best values, underline presents the second-best values.

Method	Order 1 with 15 tasks				Order 2 with 15 tasks			
	AP $\pm$ std $\uparrow$	F.Ra $\pm$ std $\downarrow$	FWT $\pm$ std $\uparrow$	BWT $\pm$ std $\uparrow$	AP $\pm$ std $\uparrow$	F.Ra $\pm$ std $\downarrow$	FWT $\pm$ std $\uparrow$	BWT $\pm$ std $\uparrow$
SeqLoRA	5.41 $\pm$ 0.54	30.73 $\pm$ 0.71	-17.32 $\pm$ 0.18	-28.54 $\pm$ 0.86	7.86 $\pm$ 0.62	35.55 $\pm$ 0.51	-9.76 $\pm$ 0.16	-30.81 $\pm$ 0.78
L2P	15.65 $\pm$ 0.45	7.12 $\pm$ 0.41	-19.86 $\pm$ 0.89	-3.33 $\pm$ 0.23	9.97 $\pm$ 0.31	17.50 $\pm$ 0.66	-16.94 $\pm$ 0.74	-12.20 $\pm$ 0.32
CodaPrompt	21.31 $\pm$ 0.24	5.12 $\pm$ 0.31	-0.86 $\pm$ 0.15	-3.31 $\pm$ 0.44	16.11 $\pm$ 0.74	12.24 $\pm$ 0.61	-7.65 $\pm$ 0.93	-10.37 $\pm$ 0.54
HidePrompt	26.45 $\pm$ 0.34	3.61 $\pm$ 0.25	-0.17 $\pm$ 0.14	-3.42 $\pm$ 0.37	25.22 $\pm$ 0.81	10.33 $\pm$ 0.33	-3.21 $\pm$ 0.52	-5.01 $\pm$ 0.23
O-LoRA	23.66 $\pm$ 0.93	29.22 $\pm$ 0.72	-0.41 $\pm$ 0.31	-24.38 $\pm$ 1.03	27.21 $\pm$ 0.88	21.25 $\pm$ 0.96	0.44 $\pm$ 0.23	-18.96 $\pm$ 0.73
InfLoRA	43.56 $\pm$ 0.63	1.45 $\pm$ 0.45	0.15 $\pm$ 0.32	-2.51 $\pm$ 0.26	41.82 $\pm$ 0.55	1.88 $\pm$ 0.13	1.14 $\pm$ 0.24	-2.71 $\pm$ 0.36
SAPT-P	42.38 $\pm$ 0.56	1.72 $\pm$ 0.34	<u>2.52</u> $\pm$ 0.44	-1.11 $\pm$ 0.27	41.55 $\pm$ 0.68	1.77 $\pm$ 0.36	1.01 $\pm$ 0.35	<u>-0.94</u> $\pm$ 0.26
SAPT-LoRA	<u>51.76</u> $\pm$ 0.71	<u>0.88</u> $\pm$ 0.11	2.21 $\pm$ 0.41	<u>-0.75</u> $\pm$ 0.25	<u>50.09</u> $\pm$ 0.36	1.76 $\pm$ 0.23	1.54 $\pm$ 0.33	-1.28 $\pm$ 0.41
SD-LoRA	45.91 $\pm$ 0.53	1.31 $\pm$ 0.26	2.11 $\pm$ 0.31	-2.13 $\pm$ 0.22	44.37 $\pm$ 0.76	<u>1.71</u> $\pm$ 0.19	<u>1.55</u> $\pm$ 0.36	-2.06 $\pm$ 0.31
CaLoRA	54.42 $\pm$ 0.55	0.25 $\pm$ 0.14	2.98 $\pm$ 0.21	0.35 $\pm$ 0.21	52.76 $\pm$ 0.67	0.85 $\pm$ 0.11	2.22 $\pm$ 0.33	0.18 $\pm$ 0.19

model-agnostic continual learning framework compatible with any transformer-based pre-trained model (e.g., T5, LLaMA-2, and ViT). Following prior works [13, 7], CaLoRA is applied to the query and value projections within the attention modules of all Transformer blocks in each pretrained model. To ensure a fair comparison with recent works on the two NLP benchmarks, we implement CaLoRA based on the pre-trained T5 (0.77B and 3B) [58] and LLaMA-2 (7B) [59] models. The ranks of LoRA are set to 4 and 8 for the LLaMA-2 and T5 models, respectively. For the ImageNet-R benchmark, following existing works [7], we use the ViT-B/16 backbone [60] supervised pre-trained on ImageNet 21K as the pre-trained model, and the rank of LoRA is set to 10. All experiments are conducted with 3 H800 and 4 V100 GPUs. More training details are provided in Appendix C.

4.2 Main Results

Results on Different Task Orders and Model Scales. Tables 2 and 3 report continual learning performance under different task orders on two NLP benchmarks. CaLoRA consistently outperforms state-of-the-art baselines, including SAPT-P, SAPT-LoRA, and SD-LoRA. Specifically, Table 2 highlights CaLoRA’s superior performance on the SuperNI benchmark. Under Order 1, it achieves an average performance gain of 2.66, with improvements of 0.46 in forward transfer and 1.1 in backward transfer, while reducing forgetting rate by 0.63. Similar gains are observed under Order 2, with increases of 2.67, 0.67, and 1.12 in average performance, forward transfer, and backward transfer, respectively, and a 0.86 reduction in forgetting rate. Table 3 shows that CaLoRA also excels on the Long Sequence benchmark. For Order 1, it improves average performance by 1.81, forward transfer by 0.83, and backward transfer by 1.44, while reducing forgetting rate by 1.05. Under Order 2, it achieves gains of 1.29, 0.88, and 1.34 in the respective metrics, with a 0.91 reduction in forgetting rate. Figure 2 further compares CaLoRA against three competitive baselines across three scaled

Table 3: Overall results (mean $\pm$ std over 3 random seeds) on Long Sequence benchmark with two task orders, evaluated using the T5-Large model. **Bold** indicates the best values, underline presents the second-best values.

Method	Order 1 with 15 tasks				Order 2 with 15 tasks			
	AP $\pm$ std $\uparrow$	F.Ra $\pm$ std $\downarrow$	FWT $\pm$ std $\uparrow$	BWT $\pm$ std $\uparrow$	AP $\pm$ std $\uparrow$	F.Ra $\pm$ std $\downarrow$	FWT $\pm$ std $\uparrow$	BWT $\pm$ std $\uparrow$
SeqLoRA	7.34 $\pm$ 0.45	80.44 $\pm$ 1.13	0.99 $\pm$ 0.24	-76.10 $\pm$ 0.97	14.11 $\pm$ 0.55	74.88 $\pm$ 0.79	0.77 $\pm$ 0.21	-71.24 $\pm$ 0.89
L2P	58.35 $\pm$ 0.71	20.31 $\pm$ 0.65	1.23 $\pm$ 0.21	-15.1 $\pm$ 0.34	58.10 $\pm$ 0.61	22.76 $\pm$ 0.42	1.55 $\pm$ 0.31	-18.34 $\pm$ 0.56
CodaPrompt	65.33 $\pm$ 0.77	11.21 $\pm$ 0.35	1.87 $\pm$ 0.26	-7.65 $\pm$ 0.46	64.43 $\pm$ 0.89	13.35 $\pm$ 0.56	1.67 $\pm$ 0.31	-10.31 $\pm$ 0.67
HidePrompt	69.78 $\pm$ 0.41	7.85 $\pm$ 0.25	2.11 $\pm$ 0.14	-4.01 $\pm$ 0.35	69.12 $\pm$ 0.86	6.43 $\pm$ 0.46	1.81 $\pm$ 0.32	-7.21 $\pm$ 0.44
O-LoRA	70.14 $\pm$ 0.64	7.87 $\pm$ 0.24	-7.10 $\pm$ 0.32	-4.04 $\pm$ 0.31	70.26 $\pm$ 0.55	5.11 $\pm$ 0.30	-7.88 $\pm$ 0.21	-5.56 $\pm$ 0.37
InfLoRA	80.26 $\pm$ 0.76	2.21 $\pm$ 0.13	0.65 $\pm$ 0.27	-3.27 $\pm$ 0.45	78.21 $\pm$ 0.64	2.99 $\pm$ 0.31	0.38 $\pm$ 0.32	-4.05 $\pm$ 0.24
SAPT-P	79.84 $\pm$ 0.45	2.45 $\pm$ 0.14	<u>3.35</u> $\pm$ 0.24	-1.45 $\pm$ 0.21	77.68 $\pm$ 0.61	2.76 $\pm$ 0.24	2.26 $\pm$ 0.29	-2.04 $\pm$ 0.26
SAPT-LoRA	<u>82.81</u> $\pm$ 0.66	<u>1.21</u> $\pm$ 0.13	2.23 $\pm$ 0.21	<u>-1.15</u> $\pm$ 0.16	<u>80.66</u> $\pm$ 0.72	<u>2.46</u> $\pm$ 0.33	2.01 $\pm$ 0.32	<u>-1.91</u> $\pm$ 0.25
SD-LoRA	81.35 $\pm$ 0.55	1.76 $\pm$ 0.32	2.51 $\pm$ 0.34	-2.71 $\pm$ 0.29	78.98 $\pm$ 0.68	3.21 $\pm$ 0.33	<u>2.33</u> $\pm$ 0.39	-2.77 $\pm$ 0.31
CaLoRA	84.62 $\pm$ 0.63	0.16 $\pm$ 0.55	4.18 $\pm$ 0.39	0.29 $\pm$ 0.13	81.95 $\pm$ 0.49	1.55 $\pm$ 0.32	3.21 $\pm$ 0.45	-0.57 $\pm$ 0.12

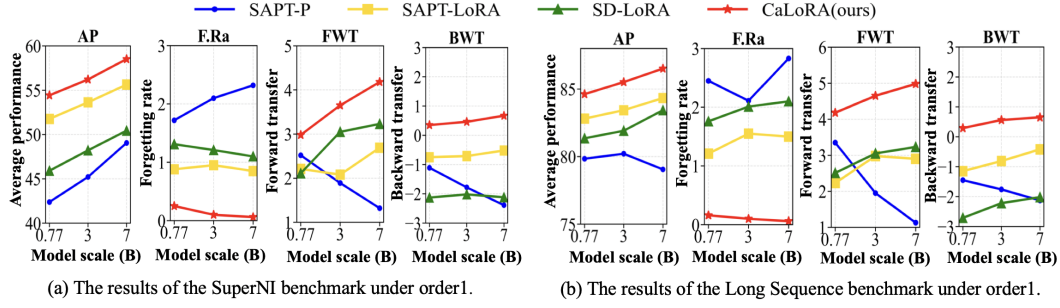


Figure 2: Comparative performance of CaLoRA against suboptimal baselines across three scaled pretrained models: T5-Large (0.77B), T5-XL (3B), and LLaMA-2 (7B).

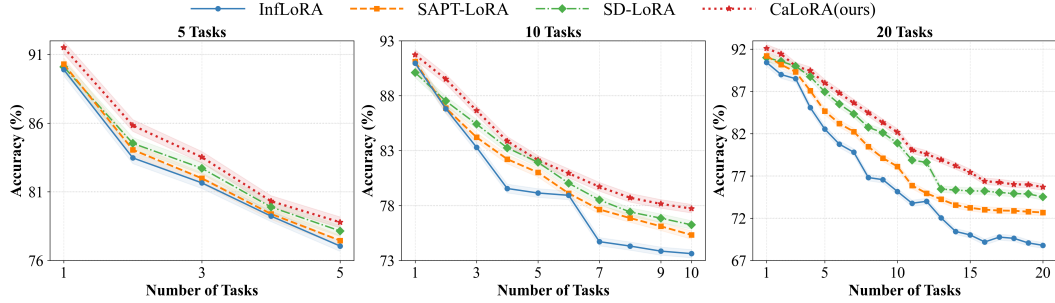


Figure 3: Variation of the performance of different methods during the learning of ImageNet-R.

pre-trained models on two NLP benchmarks in the order 1. The results demonstrate that CaLoRA consistently delivers superior continual learning performance regardless of model size.

Results across Varied Task Lengths. Table 4 presents the comparative results on the ImageNet-R benchmark for both 10-task and 20-task continual learning scenarios. CaLoRA demonstrates consistent superiority over all baseline methods in both task lengths. In the 10-task setting, CaLoRA surpasses the second-best baselines (i.e., SD-LoRA and SAPT-LoRA) with improvements: 1.48 in average performance, 0.67 in forward transfer, and 1.4 in backward transfer, while simultaneously achieving a 0.78 reduction in forgetting rate. The advantages persist in the more challenging 20-task setting, where CaLoRA maintains leads of 1.16, 0.73, and 0.97 in the respective metrics, along with a 0.57 decrease in forgetting rate. Complementing these quantitative results, Figure 3 visually demonstrates CaLoRA’s superior learning trajectory compared to three suboptimal continual learning methods on the ImageNet-R benchmark. The accuracy curves reveal that CaLoRA maintains consistently higher performance throughout the entire learning process, not just at the final evaluation stage. Additional supporting results are provided in Appendix D. In particular, these results demonstrate that CaLoRA more effectively mitigates catastrophic forgetting by enabling positive backward knowledge transfer, in comparison to several baseline methods.

Table 4: Overall results (mean $\pm$ std over 3 random seeds) on ImageNet-R (10/20 tasks) benchmark with the ViT-B/16 backbone. **Bold (underline)** indicates the best (second-best) values.

Method	10 tasks				20 tasks			
	AP $\pm$ std $\uparrow$	F.Ra $\pm$ std $\downarrow$	FWT $\pm$ std $\uparrow$	BWT $\pm$ std $\uparrow$	AP $\pm$ std $\uparrow$	F.Ra $\pm$ std $\downarrow$	FWT $\pm$ std $\uparrow$	BWT $\pm$ std $\uparrow$
SeqLoRA	62.45 $\pm$ 0.68	22.41 $\pm$ 0.58	0.86 $\pm$ 0.21	-17.61 $\pm$ 0.33	50.86 $\pm$ 0.43	32.16 $\pm$ 0.68	0.73 $\pm$ 0.13	-14.22 $\pm$ 0.37
L2P	67.66 $\pm$ 0.48	15.15 $\pm$ 0.27	1.07 $\pm$ 0.21	-13.69 $\pm$ 0.71	64.64 $\pm$ 0.41	19.81 $\pm$ 0.63	0.79 $\pm$ 0.19	-12.75 $\pm$ 0.63
CodaPrompt	71.28 $\pm$ 0.41	9.67 $\pm$ 0.32	1.71 $\pm$ 0.23	-9.32 $\pm$ 0.47	67.25 $\pm$ 0.33	11.83 $\pm$ 0.45	1.21 $\pm$ 0.17	-8.46 $\pm$ 0.54
HidePrompt	72.62 $\pm$ 0.34	8.38 $\pm$ 0.45	1.76 $\pm$ 0.23	-10.33 $\pm$ 0.57	69.88 $\pm$ 0.39	9.36 $\pm$ 0.53	1.43 $\pm$ 0.31	-8.73 $\pm$ 0.42
O-LoRA	72.15 $\pm$ 0.45	5.32 $\pm$ 0.36	-0.97 $\pm$ 0.24	-6.42 $\pm$ 0.31	66.23 $\pm$ 0.28	7.81 $\pm$ 0.63	-0.81 $\pm$ 0.17	-5.81 $\pm$ 0.37
InfLoRA	73.62 $\pm$ 0.34	4.11 $\pm$ 0.35	0.56 $\pm$ 0.23	-4.16 $\pm$ 0.43	68.79 $\pm$ 0.36	6.89 $\pm$ 0.35	0.67 $\pm$ 0.21	-4.39 $\pm$ 0.34
SAPT-P	72.82 $\pm$ 0.54	5.33 $\pm$ 0.41	2.26 $\pm$ 0.45	-2.21 $\pm$ 0.37	68.08 $\pm$ 0.35	7.95 $\pm$ 0.45	2.15 $\pm$ 0.31	-1.16 $\pm$ 0.31
SAPT-LoRA	75.32 $\pm$ 0.54	3.25 $\pm$ 0.35	3.23 $\pm$ 0.41	<u>-1.25</u> $\pm$ 0.32	72.68 $\pm$ 0.46	<u>5.25</u> $\pm$ 0.33	3.21 $\pm$ 0.21	<u>-0.89</u> $\pm$ 0.13
SD-LoRA	<u>76.24</u> $\pm$ 0.32	<u>3.11</u> $\pm$ 0.31	<u>3.51</u> $\pm$ 0.32	-1.71 $\pm$ 0.23	<u>74.52</u> $\pm$ 0.43	5.31 $\pm$ 0.35	<u>3.25</u> $\pm$ 0.26	-1.94 $\pm$ 0.23
CaLoRA	77.72 $\pm$ 0.36	2.33 $\pm$ 0.27	4.18 $\pm$ 0.33	0.15 $\pm$ 0.19	75.68 $\pm$ 0.45	4.68 $\pm$ 0.31	3.98 $\pm$ 0.23	0.08 $\pm$ 0.21

Table 5: Ablation results on SuperNI benchmark with two task orders, evaluated using the T5-Large model. **Bold** (underline) indicates the best (second-best) values.

Parameter Updating Strategy	Order 1 with 15 tasks				Order 2 with 15 tasks			
	AP $\uparrow$	F.Ra $\downarrow$	FWT $\uparrow$	BWT $\uparrow$	AP $\uparrow$	F.Ra $\downarrow$	FWT $\uparrow$	BWT $\uparrow$
w/o Task Correlation (TaC)	<u>53.15</u>	<u>0.61</u>	1.97	<u>-0.42</u>	<u>50.83</u>	<u>1.33</u>	2.01	<u>-0.61</u>
w/o Task Affinity (TaA)	51.41	0.95	1.52	-1.19	47.66	2.83	1.53	-1.26
w/ Causal Effect (CaE)	44.86	2.18	0.41	-3.45	43.72	2.83	0.51	-2.21
w/ CaE+Gradient Projection	45.05	1.81	0.93	-1.95	44.13	1.73	0.76	-1.77
w/ Gradient Projection (GradProj)	43.35	1.86	0.53	-2.12	42.16	2.33	0.21	-2.28
w/ GradProj+TaC	45.57	1.61	1.21	-1.23	44.36	1.71	1.06	-1.66
w/ GradProj+TaC+TaA (w/o CaE)	48.89	1.08	<u>2.23</u>	-0.54	47.21	1.66	<u>2.04</u>	-0.71
CaLoRA	54.42	0.25	2.98	0.35	52.76	0.85	2.22	0.18

4.3 Ablation Studies.

To evaluate the impact of each design component in CaLoRA, we conduct ablation studies using seven parameter update strategies, with the results summarized in Table 5. The ablation settings are as follows: (1) w/o Task Correlation (TaC) and (2) w/o Task Affinity (TaA) remove task correlation and affinity when optimizing the gradient, respectively; (3) w/ Causal Effect only uses the causal effect to constrain the task gradient; (4) w/ CaE+Gradient projection applies both causal effect and naïve gradient projection, without task correlation or affinity; (5) w/ Gradient Projection (GradProj) employs only naïve gradient projection, excluding causal effect, task correlation, and affinity; (6) w/ GradProj+TaC combines gradient projection and task correlation, without task affinity or causal effect; (7) w/ GradProj+TaC+TaA incorporates gradient projection, task correlation, and affinity, excluding causal effect. As shown in Table 5, the causal effect significantly improves average performance, while gradient projection, task correlation, and task affinity are essential for alleviating forgetting and enhancing both forward and backward knowledge transfer.

5 Conclusion

In this work, we go beyond the conventional focus on mitigating catastrophic forgetting and explore backward knowledge transfer in PEFT-based continual learning. To this end, we propose CaLoRA, a novel **causal-aware low-rank adaptation** framework explicitly designed to facilitate backward knowledge transfer. We introduce **parameter-level counterfactual attribution** (PaCA) to estimate the causal effect of parameters, and **cross-task gradient adaptation** (CaGA) to estimate task correlation and affinity. CaGA adaptively adjusts gradients based on these measures, enabling backward knowledge transfer without data replay. Extensive experiments on multiple benchmarks demonstrate that CaLoRA consistently outperforms existing state-of-the-art methods.

Limitations. CaLoRA has two limitations: (1) Although it avoids data replay, it requires storing old task gradients, which can lead to increased memory overhead when the number of tasks or parameter dimensionality is large. Future work will explore task correlation based on parameter function space modeling; (2) The method assumes clear task boundaries, which limits its applicability. Future work will extend CaLoRA to more general online continual learning scenarios by modeling perturbations in the space of task parameters.

Broader Impacts. In this work, we propose CaLoRA to explore backward knowledge transfer in PEFT-based continual learning. We believe that this work will have a positive social impact. By improving the efficiency and scalability of continual learning, CaLoRA has the potential to reduce the computational cost and energy consumption associated with retraining large models from scratch, thus contributing to more sustainable AI development.

6 Acknowledgment

This work was supported in part by the National Key Research and Development Program of China (Grant No. 2023YFB3107000) and the Major Key Project of PCL (Grant No. PCL2024A05).

References

- [1] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, pages 139–149, 2022.
- [2] Muhammad Gul Zain Ali Khan, Muhammad Ferjad Naeem, Luc Van Gool, Didier Stricker, Federico Tombari, and Muhammad Zeshan Afzal. Introducing language guidance in prompt-based continual learning. In *ICCV*, pages 11429–11439, 2023.
- [3] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rog rio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, pages 11909–11919, 2023.
- [4] Yu Feng, Zhen Tian, Yifan Zhu, Zongfu Han, Haoran Luo, Guangwei Zhang, and Meina Song. Cp-prompt: Composition-based cross-modal prompting for domain-incremental continual learning. *CoRR*, abs/2407.21043, 2024.
- [5] Yujie Li, Xin Yang, Hao Wang, Xiangkun Wang, and Tianrui Li. Learning to prompt knowledge transfer for open-world continual learning. In *AAAI*, pages 13700–13708, 2024.
- [6] Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *CVPR*, pages 23638–23647, 2024.
- [7] Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. S-lora: Scalable low-rank adaptation for class incremental learning. *ICLR*, 2025.
- [8] German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.
- [9] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *CVPR*, pages 23219–23230, 2024.
- [10] Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. SAPT: A shared attention framework for parameter-efficient continual learning of large language models. In *ACL*, pages 11641–11661, 2024.
- [11] Xinyuan Gao, Songlin Dong, Yuhang He, Qiang Wang, and Yihong Gong. Beyond prompt learning: Continual adapter for efficient rehearsal-free continual learning. *CoRR*, abs/2407.10281, 2024.
- [12] Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 10658–10671. Association for Computational Linguistics, 2023.
- [13] Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. SAPT: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, *ACL*, pages 11641–11661. Association for Computational Linguistics, 2024.
- [14] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *AISTATS*, pages 3762–3773, 2020.
- [15] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. TRGP: trust region gradient projection for continual learning. In *ICLR*, 2022.
- [16] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Beyond not-forgetting: Continual learning with backward knowledge transfer. In *Advances in Neural Information Processing Systems*, volume 35, pages 16165–16177. Curran Associates, Inc., 2022.

- [17] Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 4133–4145. Association for Computational Linguistics, 2023.
- [18] Shwai He, Liang Ding, Daize Dong, Jeremy Zhang, and Dacheng Tao. Sparseadapter: An easy approach for improving the parameter-efficiency of adapters. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 2184–2190. Association for Computational Linguistics, 2022.
- [19] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.
- [20] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogério Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, pages 11909–11919, 2023.
- [21] Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems*, 2023.
- [22] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.
- [23] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *NeurIPS*, pages 1022–1035, 2021.
- [24] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *ICML*, pages 2790–2799, 2019.
- [25] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, pages 3045–3059, 2021.
- [26] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL/IJCNLP*, pages 4582–4597, 2021.
- [27] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 7319–7328. Association for Computational Linguistics, 2021.
- [28] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *TPAMI*, 46(8):5362–5383, 2024.
- [29] Rui Wang, Wenhua Li, Kaili Shen, Tao Zhang, and Xiangke Liao. Evolutionary multi-tasking optimization for high-efficiency time series data clustering. *Tsinghua Science and Technology*, 29(2):343–355, 2024.
- [30] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, pages 11816–11825, 2019.
- [31] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *NeurIPS*, pages 11849–11860, 2019.
- [32] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *ICML*, pages 1952–1961, 2020.

- [33] Qing Sun, Fan Lyu, Fanhua Shang, Wei Feng, and Liang Wan. Exploring example influence in continual learning. In *NeurIPS*, 2022.
- [34] Yan-Shuo Liang and Wu-Jun Li. Loss decoupling for task-agnostic continual learning. In *NeurIPS*, 2023.
- [35] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995, 2017.
- [36] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 144–161, 2018.
- [37] Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. Continual learning with node-importance based adaptive group sparse regularization. In *NeurIPS*, 2020.
- [38] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *ICML*, pages 3925–3934, 2019.
- [39] Steven C. Y. Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. In *NeurIPS*, pages 13647–13657, 2019.
- [40] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, pages 631–648, 2022.
- [41] Michael Mozer and Paul Smolensky. Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *NeurPS*, pages 107–115, 1988.
- [42] Ehud D. Karnin. A simple procedure for pruning back-propagation trained neural networks. *Trans. Neural Networks*, 1(2):239–242, 1990.
- [43] Mohamed Elsayed and A. Rupam Mahmood. Addressing loss of plasticity and catastrophic forgetting in continual learning. In *ICLR*. OpenReview.net, 2024.
- [44] Patrick Schwab, Djordje Miladinovic, and Walter Karlen. Granger-causal attentive mixtures of experts: Learning important features with neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 4846–4853, 2019.
- [45] Patrick Schwab and Walter Karlen. Explain: Causal explanations for model interpretation under uncertainty. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, pages 10220–10230, 2019.
- [46] Chaoyang Li, Jianyang Qin, Jinhao Cui, Zeyu Liu, Ning Hu, and Qing Liao. Joint scheduling of causal prompts and tasks for multi-task learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 25124–25134, 2025.
- [47] Mohamed Elsayed and A. Rupam Mahmood. Addressing loss of plasticity and catastrophic forgetting in continual learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- [48] Mohamed Elsayed and A. Rupam Mahmood. Hessscale: Scalable computation of hessian diagonals. *CoRR*, abs/2210.11639, 2022.
- [49] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *ICLR*, 2021.
- [50] Chaoyang Li, Heyan Chai, Yan Jia, Ning Hu, and Qing Liao. Multitask causal contrastive learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(9):17322–17335, 2025.

- [51] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 5085–5109. Association for Computational Linguistics, 2022.
- [52] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabza, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations, ICLR*. OpenReview.net, 2023.
- [53] Matteo Boschini, Lorenzo Bonicelli, Angelo Porrello, Giovanni Bellitto, Matteo Pennisi, Simone Palazzo, Concetto Spampinato, and Simone Calderara. Transfer without forgetting. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXIII*, volume 13683 of *Lecture Notes in Computer Science*, pages 692–709. Springer, 2022.
- [54] Chin Yew Lin. Rouge: A package for automatic evaluation of summaries. 2004.
- [55] Arslan Chaudhry, Puneet Kumar Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Computer Vision - ECCV*, volume 11215 of *Lecture Notes in Computer Science*, pages 556–572. Springer, 2018.
- [56] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 6467–6476, 2017.
- [57] Zixuan Ke and Bing Liu. Continual learning of natural language processing tasks: A survey. *CoRR*, abs/2211.12701, 2022.
- [58] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [59] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabza, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [60] 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 *9th International Conference on Learning Representations, ICLR*. OpenReview.net, 2021.
- [61] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*,., pages 248–255. IEEE Computer Society, 2009.

- [62] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 8024–8035, 2019.
- [63] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP*, pages 38–45. Association for Computational Linguistics, 2020.
- [64] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR*. OpenReview.net, 2019.
- [65] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.

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: The abstract and Section 1 have claimed the contributions and scope in the paper.

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 the 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: [\[NA\]](#)

Justification: This paper does not include theoretical results.

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: The framework design have been fully described in the Section 3 and the training details are provided in the subsection 4.1 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: The key code can be found in the supplementary 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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Please refer to the subsection 4.1 and Appendix B and 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: Please refer to main results in the subsection 4.2.

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 the subsection 4.1 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: We have reviewed and the research conforms with the NeurIPS Code of Ethics.

Guidelines:

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

10. Broader impacts

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

Answer: [Yes]

Justification: Please refer to section 5.

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: The paper does not pose such risks.

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: All creators of datasets are properly credited by citations.

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: [Yes]

Justification: The paper provides a detailed description of the use of T5 and Llama2 as pre-trained models in the subsection 4.1.

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 Singular Value Decomposition

Let $\mathbf{G} \in \mathbb{R}^{d_O \times d_I}$ denote a matrix (in our case, the gradient matrix of a task). The singular value decomposition of $\mathbf{G}$ is:

$$\mathbf{G} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad (10)$$

where $\mathbf{U} \in \mathbb{R}^{d_O \times d_O}$ contains the left singular vectors (column-orthonormal), $\mathbf{V} \in \mathbb{R}^{d_I \times d_I}$ contains the right singular vectors, $\mathbf{\Sigma} \in \mathbb{R}^{d_O \times d_I}$ is a diagonal matrix whose diagonal entries are the singular values $\{\sigma_1, \sigma_2, \dots, \sigma_r\}$, with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, and $r = \text{rank}(\mathbf{G})$.

For a given task $\mathcal{T}_k$, let $\mathbf{G}_{\mathcal{T}_k} \in \mathbb{R}^{d_O \times d_I}$ denote the task gradient matrix. Applying singular value decomposition (SVD) to $\mathbf{G}_{\mathcal{T}_k}$, we obtain:

$$\mathbf{G}_{\mathcal{T}_k} = \mathbf{U}_{\mathcal{T}_k} \mathbf{\Sigma}_{\mathcal{T}_k} \mathbf{V}_{\mathcal{T}_k}^\top. \quad (11)$$

We then extract the top- r left singular vectors $\mathbf{U}_{\mathcal{T}_k} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r] \in \mathbb{R}^{d_O \times r}$, corresponding to the r largest singular values, which capture the dominant directions of the gradient space. These vectors form an orthonormal basis that spans the input subspace relevant to task $\mathcal{T}_k$.

This gradient-based subspace projection method provides a theoretically grounded and computationally tractable method for quantifying task correlation [6, 15, 16]. By leveraging SVD, we extract meaningful input subspaces. By measuring the normalized projection of new gradients onto old subspaces, we obtain a reasonable task correlation that facilitates the analysis of backward knowledge transfer among tasks.

B Additional Dataset Details

Two NLP Benchmarks. Following previous work [13], we adopt SuperNI and Long Sequence as the NLP benchmarks to evaluate continual learning methods for large language models (LLMs). These benchmarks’ detailed descriptions and evaluation metrics are presented in Table 6. We consider two different task orders for each benchmark, as shown in Table 7. Specifically, the SuperNI Benchmark [51] comprises a diverse set of NLP tasks, each accompanied by expert-written instructions, facilitating rigorous and realistic evaluation in continual learning settings. It consists of 15 sequential tasks. For each task, 1,000 instances are randomly sampled for training, and 100 instances are used for validation and testing. The Long Sequence Benchmark [52] also includes 15 classification tasks, specifically designed for continual learning with LLMs. For each task, 1,000 training samples are randomly selected, and 500 samples per class are used for both validation and testing. This benchmark emphasizes challenges in handling long-context dependencies and task diversity in a sequential learning setting.

Table 6: The details of two NLP benchmarks.

SuperNI Benchmark			Long Sequence Benchmark		
Dataset name	Task	Metric	Dataset name	Task	Metric
1. task639	dialogue generation	Rouge-L	1. Yelp	sentiment analysis	accuracy
2. task1590	dialogue generation	Rouge-L	2. Amazon	sentiment analysis	accuracy
3. task1729	dialogue generation	Rouge-L	3. DBpedia	topic classification	accuracy
4. task181	information extraction	Rouge-L	4. Yahoo	topic classification	accuracy
5. task748	information extraction	Rouge-L	5. AG News	topic classification	accuracy
6. task1510	information extraction	Rouge-L	6. MNLI	natural language inference	accuracy
7. task002	question answering	Rouge-L	7. QQP	paragraph detection	accuracy
8. task073	question answering	Rouge-L	8. RTE	natural language inference	accuracy
9. task591	question answering	Rouge-L	9. SST-2	sentiment analysis	accuracy
10. task511	summarization	Rouge-L	10. WiC	word sense disambiguation	accuracy
11. task1290	summarization	Rouge-L	11. CB	natural language inference	accuracy
12. task1572	summarization	Rouge-L	12. COPA	question and answering	accuracy
13. task363	sentiment analysis	accuracy	13. BoolQA	boolean question and answering	accuracy
14. task875	sentiment analysis	accuracy	14. MultiRC	question and answering	accuracy
15. task1687	sentiment analysis	accuracy	15. IMDB	sentiment analysis	accuracy

Table 7: Different task orders of two NLP benchmarks.

Order	SuperNI Benchmark	Long Sequence Benchmark
1	task1572 → task363 → task1290 → task181 → task002 → task1510 → task639 → task1729 → task073 → task1590 → task748 → task511 → task591 → task1687 → task875	MNLI → CB → WiC → COPA → QQP → BoolQA → RTE → IMDB → Yelp → Amazon → SST-2 → DBpedia → AG News → MultiRC → Yahoo
2	task748 → task073 → task1590 → task639 → task1572 → task1687 → task591 → task363 → task1510 → task1729 → task181 → task511 → task002 → task1290 → task875	Yelp → Amazon → MNLI → CB → COPA → QQP → RTE → IMDB → SST-2 → DBpedia → AG News → Yahoo → MultiRC → BoolQA → WiC

The CV Benchmark. We adopt the ImageNet-R dataset [53], which consists of 200 ImageNet [61] classes rendered through various artistic styles. Introduced into the continual learning by prior works [6, 7], ImageNet-R has become a widely used benchmark for evaluating parameter-efficient fine-tuning (PEFT) methods. Following prior works [6, 7, 20], we partition the dataset into 5, 10, and 20 tasks, corresponding to 40, 20, and 10 classes per task, respectively. These configurations allow for a comprehensive assessment of the scalability and adaptability of continual learning methods across varying levels of task granularity.

C Additional Baselines and Training Details

Baselines. We evaluate our proposed CaLoRA against nine PEFT-based continual learning baselines: (1) SeqLoRA sequentially trains LoRA modules following the task order; (2) L2P [1] dynamically selects and updates prompts from a fixed prompt pool based on the input; (3) CodaPrompt [20] composes prompts dynamically via attention mechanisms to mitigate forgetting and improve adaptability; (4) HidePrompt [21] introduces a hierarchical decomposition framework to enhance prompt-based continual learning; (5) O-LoRA [12] learns each task in an orthogonal LoRA subspace and aggregates LoRA weights at inference time; (6) InfLoRA [6] constructs task-specific LoRA subspaces to reduce interference between previously learned and new tasks; (7) SAPT-P (SAPT-Prompt) and (8) SAPT-LoRA are two variants of SAPT [13], which employ a shared attentive learning and selection module to align and select appropriate PEFT strategies based on prompt tuning and LoRA, respectively; (9) SD-LoRA [7] incrementally decouples the learning of direction and magnitude in LoRA parameters.

Training Details. All experiments are implemented using PyTorch [62] and the Transformers library [63]. Specifically, for the two NLP benchmarks, we use the AdamW [64] optimizer in T5 and LLaMA-2 with learning rates of 0.0003 and 0.0005, respectively. The experiments are conducted on three H800 GPUs to enhance computational efficiency, with per-GPU batch sizes set to 2 for LLaMA-2 and 16 for T5. For the SuperNI benchmark, the number of training epochs is set to 100 for T5 and 50 for LLaMA-2. For the LongSequence benchmark, the number of training epochs is set to 10 for T5 and 20 for LLaMA-2. For the ImageNet-R benchmark, we follow prior work [6] and use the Adam optimizer [65] with a learning rate of 0.0005. Each experiment is conducted on a single H800 or V100 GPU, with the batch size uniformly set to 128. Each task is trained for 50 epochs.

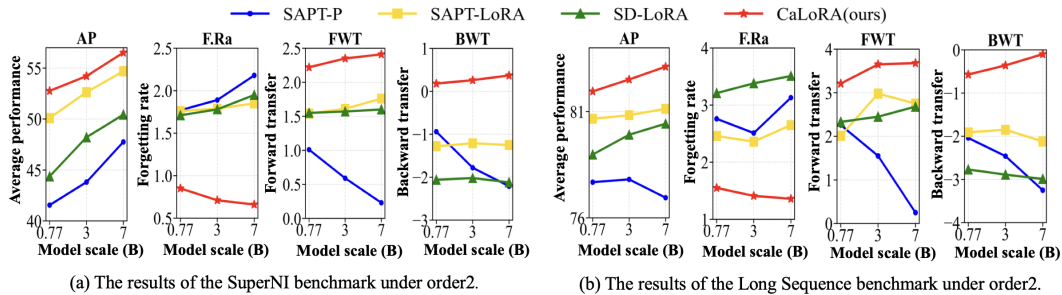


Figure 4: Comparative performance of CaLoRA against suboptimal baselines across three scaled pretrained models (T5-Large with 0.77B parameters, T5-XL with 3B parameters, and LLaMA-2 with 7B parameters) on two NLP benchmarks under order 2.

Table 8: Overall results (mean $\pm$ std over 3 random seeds) on ImageNet-R (5 tasks) with the ViT-B/16 backbone. **Bold** (underline) indicates the best (second-best) values.

Method	5 tasks			
	AP $\pm$ std $\uparrow$	F.Ra $\pm$ std $\downarrow$	FWT $\pm$ std $\uparrow$	BWT $\pm$ std $\uparrow$
SeqLoRA	63.15 $\pm$ 0.51	21.05 $\pm$ 0.43	1.34 $\pm$ 0.21	-15.11 $\pm$ 0.24
L2P	68.99 $\pm$ 0.39	13.44 $\pm$ 0.31	1.86 $\pm$ 0.25	-11.43 $\pm$ 0.56
CodaPrompt	74.88 $\pm$ 0.45	8.76 $\pm$ 0.24	2.43 $\pm$ 0.23	-8.02 $\pm$ 0.36
HidePrompt	74.46 $\pm$ 0.41	8.01 $\pm$ 0.33	2.76 $\pm$ 0.26	-8.33 $\pm$ 0.31
O-LoRA	74.75 $\pm$ 0.32	5.21 $\pm$ 0.29	-0.16 $\pm$ 0.21	-5.11 $\pm$ 0.28
InfLoRA	75.82 $\pm$ 0.41	3.78 $\pm$ 0.35	0.98 $\pm$ 0.34	-3.88 $\pm$ 0.35
SAPT-P	75.01 $\pm$ 0.52	3.41 $\pm$ 0.34	3.41 $\pm$ 0.21	-1.76 $\pm$ 0.28
SAPT-LoRA	77.13 $\pm$ 0.34	2.82 $\pm$ 0.31	<u>4.56</u> $\pm$ 0.41	<u>-0.82</u> $\pm$ 0.19
SD-LoRA	<u>78.05</u> $\pm$ 0.46	<u>2.42</u> $\pm$ 0.33	4.45 $\pm$ 0.41	-1.25 $\pm$ 0.16
CaLoRA	79.11 $\pm$ 0.39	1.53 $\pm$ 0.25	4.98 $\pm$ 0.34	0.22 $\pm$ 0.19

Table 9: Comparison on ImageNet-R (20 tasks) in terms of computation (GFLOPs), parameters, and storage efficiency. $\dagger$ indicates that the results are from SD-LoRA [7].

Method	GFLOPs	Learnable Parameters (M)	Stored Features (M)
L2P $\dagger$	70.14	0.48	0
CodaPrompt $\dagger$	70.61	0.38	0
HidePrompt $\dagger$	70.36	0.08	0.15
O-LoRA	35.12	0.19	0
InfLoRA $\dagger$	35.12	0.37	0.10
SAPT-P	77.11	0.34	0.15
SAPT-LoRA	35.12	0.37	0.15
SD-LoRA $\dagger$	35.12	0.37	0
CaLoRA	35.12	0.37	0.10

D Additional Results

Additional Results of NLP and CV Benchmarks. Figure 4 presents a comparison between CaLoRA and three strong baselines across three different scales of pre-trained models on two NLP benchmarks under task order 2. The results show that CaLoRA consistently achieves superior continual learning performance across all model sizes. Table 8 presents results on five tasks from the ImageNet-R benchmark. CaLoRA achieves the highest performance among all compared methods under the continual learning setting. These results support the applicability of CaLoRA across both NLP and vision tasks, demonstrating its generality in diverse continual learning scenarios.

Analysis of Computation, Parameter, and Storage Efficiency. As shown in Table 9, we compare the inference efficiency of various PEFT-based continual learning methods across three dimensions: GFLOPs, the number of trainable parameters, and feature storage requirements. Table 9 demonstrates that LoRA-based methods generally achieve superior inference efficiency. Compared with most state-of-the-art (SOTA) methods, our proposed CaLoRA does not significantly increase parameter or computational overhead while effectively improving continual learning performance.