Data Efficient Adaptation in Large Language Models via Continuous Low-Rank Fine-Tuning

Xiao Han*
Zhejiang University of Technology,
Zhejiang Key Laboratory of
Visual Information Intelligent Processing
Hangzhou, China
hahahenha@gmail.com

Zimo Zhao*
City University of Hong Kong
Hong Kong, China
zmzhao6-c@my.cityu.edu.hk

Wanyu Wang*
City University of Hong Kong
Hong Kong, China
wanyuwang4-c@my.cityu.edu.hk

Maolin Wang
City University of Hong Kong
Hong Kong, China
Morin.wang@my.cityu.edu.hk

Zitao Liu
Jinan University
Jinan, China
liuzitao@jnu.edu.cn

Yi Chang
Jilin University
Jilin, China
yichang@jlu.edu.cn

Xiangyu Zhao†
City University of Hong Kong
Hong Kong, China
xianzhao@cityu.edu.hk

Abstract

Recent advancements in Large Language Models (LLMs) have emphasized the critical role of fine-tuning (FT) techniques in adapting LLMs to specific tasks, especially when retraining from scratch is computationally infeasible. Fine-tuning enables LLMs to leverage task- or domain-specific data, producing models that more effectively meet the requirements of targeted applications. However, conventional FT approaches often suffer from catastrophic forgetting and suboptimal data efficiency, limiting their real-world applicability. To address these challenges, this paper proposes **DEAL**, a novel framework that integrates Low-Rank Adaptation (LoRA) with a continuous fine-tuning strategy. By incorporating knowledge retention and adaptive parameter update modules, the framework mitigates the limitations of existing FT methods while maintaining efficiency. Experiments on **15** diverse datasets show that **DEAL** consistently outperforms baseline methods, yielding substantial gains in task accuracy and resource efficiency. These findings demonstrate the potential of our approach to advance continual adaptation in LLMs by enhancing task performance while improving resource efficiency. The source code is publicly available at https://github.com/Applied-Machine-Learning-Lab/DEAL.

1 Introduction

The advent of Large Language Models (LLMs) has catalyzed transformative advances in Natural Language Processing (NLP), enabling breakthroughs across healthcare, education, web technologies, and other domains [1–5]. However, training and utilizing these models to evolve to real-world

*Equal contribution.
†Corresponding author.

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

demands remains a critical challenge. Direct fine-tuning of billion-scale-parameter models incurs prohibitive computational costs, creating accessibility barriers for resource-constrained researchers and institutions. Even for small and medium-sized enterprises, it is also difficult to independently implement pre-training of these models. In such scenarios, Parameter-Efficient Fine-Tuning (PEFT) methods—particularly Low-Rank Adaptation (LoRA)—have emerged as practical solutions that leverage pre-trained models by "standing on the shoulders of giants" [6, 7]. By applying low-rank matrix decompositions, LoRA reduces the number of trainable parameters by over 90% while maintaining baseline performance. By selectively updating task-specific subspaces, LoRA enables targeted knowledge integration in non-stationary environments—a capability aligned with the core objectives of continual learning.

As LLMs require ongoing updates to ensure their knowledge remains current and relevant over time, continual learning of LoRA is required to integrate new information while preserving existing capabilities [8]. This approach mitigates catastrophic forgetting by freezing most parameters and restricting updates to a low-rank matrix, allowing LoRA to differentially activate specific knowledge within the model. For instance, a LoRA-based model trained on Wiki-QA [9] can be further refined on TruthfulQA [10] to enhance performance. However, while continual learning of the LoRA module only unlocks up-to-date task-specific outcomes, it may compromise cross-domain performance [11], particularly when smaller, specialized datasets lack the breadth of the original pre-training data [12]. Therefore, could we design a high-level fine-tuning method that maintains excellent performance across all tasks while allowing continuous fine-tuning with small-scale datasets?

Several existing studies have explored this problem. In particular, continuous learning can be achieved through two main strategies: (1) direct model editing, and (2) introducing additional adapters. On the one hand, studies [13–18] have shown that key-value-like structures in the Transformer layers can be directly edited. For example, ROME [13] and MEMIT [14] directly update the key-value-like structures in Transformer layers via causal weight interventions. However, these approaches require massive additional experiments to pinpoint which neurons to edit, making it both inefficient and costly. On the other hand, for LoRA-based LLMs, the process of locating and modifying parameters in the low-rank matrix remains largely opaque [19]. To reduce the complexity of targeting specific parameters, many studies turn to stack additional adapter modules [20–23]. Yet, these modules inevitably impose extra computational overhead. Consequently, interpretability and efficiency remain the two major challenges in applying continuous learning to LoRA-tuned LLMs.

To address these limitations, we introduce **D**ata-**E**fficient **A**daptation via continuous **L**ow-rank fine-tuning (**DEAL**), a method that facilitates efficient knowledge acquisition while preserving the interpretability of model updates. Specifically, we design a wavelet kernel to adaptively preserve core features of historical knowledge in the filtered low-rank matrix while seamlessly incorporating new information. By focusing on core aspects of historical knowledge, DEAL prevents catastrophic forgetting, thus maintaining model performance across multiple tasks. This innovative approach provides a robust framework for continuous learning, making it an effective solution for dynamic data environments. Our contributions of the paper are summarized as follows:

1. We introduce DEAL, an innovative continual learning framework that efficiently utilizes small amounts of new data for continuous learning, thereby avoiding the need for relearning and significantly conserving computing resources.

2. We leverage a wavelet kernel to preserve historical knowledge and deploy differentiated regularization terms to control the knowledge updating process, improving both transparency and efficiency. Additionally, by simply replacing the original low-rank matrices with their fine-tuned counterparts, DEAL ensures that inference time remains unchanged.

3. Comprehensive experiments on 15 multi-task open-source datasets validate the effectiveness and efficiency of our framework. These experiments demonstrate its ability to maintain high performance across different tasks while efficiently managing computational resources.

2 Preliminaries

In this section, we give the definitions of LoRA-based LLM at first and then introduce the continual learning for LoRA fine-tuning. Finally, we state the problem that we solve in this paper.

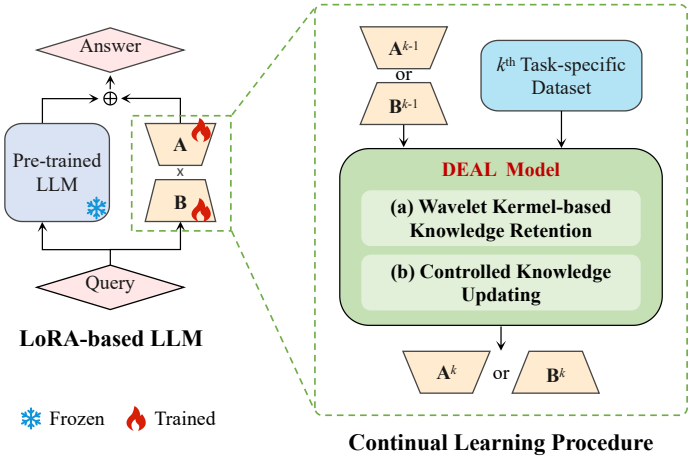

Figure 1: The framework overview.

LoRA-based LLM. LoRA enhances LLM by introducing low-rank matrices to weight updates during fine-tuning. Given a pre-trained weight matrix $\boldsymbol{W} \in \mathbb{R}^{m \times n}$, LoRA decomposes the weight update $\Delta \boldsymbol{W}$ into two smaller matrices $\boldsymbol{A} \in \mathbb{R}^{m \times r}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times r}$, where r is a user-defined rank ($r << \min\{m, n\}$). The update is then expressed as $\Delta \boldsymbol{W} = \boldsymbol{A} \times \boldsymbol{B}^\top$, allowing the model to adapt to new tasks with significantly fewer parameters compared to full fine-tuning. This decomposition reduces the number of trainable parameters from $m \times n$ to $m \times r + n \times r$, leading to substantial computational savings.

Continual LoRA Fine-Tuning. The parameter-efficient fine-tuning approaches of LoRA enable LLMs to learn new tasks while retaining performance on previously learned tasks. By updating only the low-rank matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, the LoRA-based LLM mitigates catastrophic forgetting, a common challenge in continual learning, by enabling the model to learn new tasks while retaining performance on prior tasks.

In this paper, we aim to effectively learn new tasks with acceptable training cost, while retaining performance on previously learned tasks. Mathematically, this involves adjusting the weight matrix $\boldsymbol{W}$ of the model $\mathcal{A}_{\boldsymbol{W}}$ by introducing low-rank updates $\Delta \boldsymbol{W} = \boldsymbol{A} \times \boldsymbol{B}^\top$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are low-rank matrices. The learning target is to minimize the loss function $\mathcal{L}$ over the new task data $\mathcal{D}_{\text{new}}$, subject to a regularization term that penalizes changes to the original parameters to prevent catastrophic forgetting. This can be formulated as:

$$\min_{\boldsymbol{A},\boldsymbol{B}} \mathcal{L}\left(\mathcal{A}_{\boldsymbol{W}+\boldsymbol{A}\times\boldsymbol{B}^\top}, \mathcal{D}_{\text{new}}\right) + \lambda |\boldsymbol{A} \times \boldsymbol{B}^\top|, \tag{1}$$

where λ is a hyperparameter controlling the trade-off between learning new information and retaining prior knowledge. The goal is to find the optimal low-rank matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ that allow the model to adapt to new tasks while preserving its performance on previous tasks.

3 Methodology

In this section, we first provide a framework overview of DEAL and then we introduce each part of DEAL in detail. Finally, we outline the training procedure, which fine-tunes the low-rank matrices to ensure the model adapts to new tasks without sacrificing performance on previously learned tasks.

3.1 Framework Overview

Figure 1 shows the overall framework for continuous learning on domain-specific dataset, which consists of a wavelet kernel-based knowledge retention module and a controlled knowledge updating module. **(a) Wavelet Kernel-based Knowledge Retention Module** extracts and filters singular values from the low-rank matrices to preserve the core representations of historical knowledge, which should be maintained throughout continual learning. **(b) Controlled Knowledge Updating Module**

applies higher-order regularization to constrain parameter updates in LoRA, thereby regulating the integration of new knowledge while minimizing disruption to previously learned representations. In LoRA, the small-parameter knowledge representation embedded in the low-rank matrix could activate the corresponding understanding and reasoning abilities in the original LLM, facilitating the learning of new tasks. These two modules allow the model to effectively learn and incorporate new information without disrupting previously acquired knowledge. As for the inference phase, the updated low-rank matrix, with its new knowledge representation, will directly replace the corresponding part in the original LoRA module, ensuring that the inference delay of DEAL remains unaffected.

3.2 Wavelet Kernel-based Knowledge Retention

Due to limited hardware resources, continual learning with LoRA-based LLM typically involves fine-tuning only the LoRA module, while the pre-trained LLM parameters remain unchanged. However, updates to the low-rank matrix can increasingly hinder the model's ability to retain the original features crucial for activating the corresponding capabilities within the LLM. To address this, we introduce a wavelet kernel to filter and preserve the key features of LoRA during continual learning.

In the LoRA module, the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are singular, meaning they are not full-rank. This presents a challenge for effective feature extraction. We assume that the singular matrix $\boldsymbol{Y} := \boldsymbol{A}$ or $\boldsymbol{B}$ can be decomposed into a task-relevant component $\boldsymbol{X}$ and a redundant or noisy component $\boldsymbol{D}$, i.e., $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{D}$.

Here, $\boldsymbol{X}$ denotes the core feature matrix, capturing the intrinsic low-rank structure that encodes task-relevant semantics. According to the Eckart—Young—Mirsky theorem [24], the best low-rank approximation of a matrix in the Frobenius norm sense is achieved via truncation of its singular value decomposition (SVD). This motivates our use of truncated SVD to estimate $\boldsymbol{X}$ from the observed matrix $\boldsymbol{Y}$, with the goal of recovering task-relevant features from its singular representation.

We further assume that both $\boldsymbol{Y}$ and $\boldsymbol{X}$ lie in $\mathbb{R}^{n\times r}$, sharing the same dimensions. Their corresponding singular value decompositions (SVD) can be given by:

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{S}_1 & 0 \\ 0 & \boldsymbol{S}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{Q}_1^\top \boldsymbol{V}_{x1}^H \\ \boldsymbol{Q}_2^H \boldsymbol{V}_{x2}^\top \end{pmatrix}, \tag{2}$$

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{U}_x \boldsymbol{\Sigma}_x \boldsymbol{V}_x \\ &= \begin{pmatrix} \boldsymbol{U}_{x1} & \boldsymbol{U}_{x2} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma}_{x1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{V}_{x1} \\ \boldsymbol{V}_{x2} \end{pmatrix}, \end{aligned} \tag{3}$$

where $\boldsymbol{U}_{x1} \in \mathbb{R}^{n_x \times r_x}$, $\boldsymbol{U}_{x1} \in \mathbb{R}^{n_x \times (n_x - r_x)}$, $\boldsymbol{V}_{x1} \in \mathbb{R}^{r_x \times r}$, $\boldsymbol{V}_{x2} \in \mathbb{R}^{(n_x - r_x)\times r}$.

The following theorem states that we cannot directly compute the core features of $\boldsymbol{X}$ from $\boldsymbol{Y}$:

Theorem 1 *Let $\boldsymbol{Y}$ be the observed data matrix and $\boldsymbol{X}$ the underlying core feature matrix. Then, without additional constraints, there does not exist a pair of matrices $\boldsymbol{P}_1$ and $\boldsymbol{U}_{x1}$ such that $\boldsymbol{P}_1 = \boldsymbol{U}_{x1}$. See Appendix A.2.*

Therefore, we aim to recover the core feature matrix X by representing it as a linear combination of the columns of the observed matrix Y. To this end, we introduce a coefficient matrix H and formulate the following least-squares objective:

$$\min_{\boldsymbol{H}} \|\boldsymbol{Y}\boldsymbol{H} - \boldsymbol{X}\|_F^2, \tag{4}$$

where $||\cdot||_F$ denotes the Frobenius norm. In this formulation, $\boldsymbol{X}$ is treated as the target (e.g., the ideal low-rank component), and $\boldsymbol{H}$ is the optimization variable that linearly combines the basis vectors in $\boldsymbol{Y}$ to approximate $\boldsymbol{X}$. The optimal $\boldsymbol{H}$ can be derived as:

$$\boldsymbol{H} = \left(\boldsymbol{Y}^\top \boldsymbol{Y}\right)^{-1} \boldsymbol{Y}^\top \boldsymbol{X}. \tag{5}$$

Then, $\hat{\boldsymbol{X}}$, the minimum variance estimate of $\boldsymbol{X}$, can be presented as:

$$\hat{\boldsymbol{X}} = \boldsymbol{Y}\boldsymbol{H} = \boldsymbol{Y}\left(\boldsymbol{Y}^\top \boldsymbol{Y}\right)^{-1} \boldsymbol{Y}^\top \boldsymbol{X}, \tag{6}$$

Obviously, we cannot calculate $\hat{\boldsymbol{X}}$ directly by Eq. (6). Note that $\boldsymbol{P_Y} = \boldsymbol{Y}\left(\boldsymbol{Y}^\top \boldsymbol{Y}\right)^{-1}\boldsymbol{Y}^\top$ is the orthogonal projection operator onto the column space of $\boldsymbol{Y}$. Importantly, $\hat{\boldsymbol{X}}$ is denoised when the target $\boldsymbol{X}$ lies entirely within that subspace. To proceed with estimation, we assume that the redundant features in the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ behave as white noise, i.e., $\boldsymbol{D}^\top \boldsymbol{D} = \sigma_{\boldsymbol{D}}^2 I$, $\boldsymbol{X}^\top \boldsymbol{D} = 0$, where $\sigma_{\boldsymbol{D}}^2$ is the variance of the noise. Then we can simply $\hat{\boldsymbol{X}}$:

$$\hat{\boldsymbol{X}} = \sum_{k=1}^{r_x} \frac{\sigma_k^2 - \sigma_{\boldsymbol{D}}^2}{\sigma_k} \boldsymbol{u}_k \boldsymbol{v}_k^\top, \tag{7}$$

where $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ is the left and right singular vectors of $\boldsymbol{Y}$, σ_k is the k-th largest singular value of $\boldsymbol{Y}$ ($\sigma_1 > \sigma_2 > \cdots > \sigma_r$). In traditional signal analysis algorithms, large singular values represent low-frequency data distribution and macro trends, and small singular values represent high-frequency disturbances [25]. However, since σ^2 in the above formula is unknown, we will define a series of wavelet functions at different scales for feature filtering. Here, we use the heat kernel as a low-pass filter:

$$\phi_{\sigma_j^2, c_j}(\boldsymbol{X}) = \exp\left(-\frac{1}{2\sigma_j^2}||\boldsymbol{X} - c_j||^2\right), \tag{8}$$

where c_j is the center of the j-th kernel, σ_j^2 represents the width of the kernel. By setting a series of different heat kernel widths $\boldsymbol{\sigma^2} = [\sigma_1^2, \sigma_2^2, \cdots]^\top$, defining a series of learnable diagonal matrices $\boldsymbol{g} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots]^\top$ and learnable centers $\boldsymbol{C} = [c_1, c_2, \cdots]^\top$, we can define a wavelet neural network to extract the features of $\hat{\boldsymbol{X}}$ from $\boldsymbol{Y}$:

$$\boldsymbol{H}_{:,i}^{k+1} = \delta\left(\sum_j \phi_{\sigma_j^2, c_j} \boldsymbol{g}_j \phi^{-1}_{\sigma_j^2, c_j} \boldsymbol{H}_{:,j}^k\right), \tag{9}$$

where $\delta(\cdot)$ is the activation function, $\boldsymbol{H}^0 := \boldsymbol{Y}$, and $\hat{\boldsymbol{X}} = \boldsymbol{H}^K$. Here K is the total number of layers and $k \in \{0, 1, \cdots, K-1\}$. As ϕ is the heat kernel, i.e. $\phi^{-1}(x) = \phi(-x)$), we could simply Eq. (9) to be:

$$\boldsymbol{H}_{:,i}^{k+1} = \delta\left(\sum_j \phi_{\sigma_j^2, c_j} \boldsymbol{g}_j \phi_{-\sigma_j^2, c_j} \boldsymbol{H}_{:,j}^k\right). \tag{10}$$

Eq. (10) avoids directly calculating the inverse of the function and improves calculation efficiency.

3.3 Controlled Knowledge Updating

In the previous section, we derive the core features of the original knowledge. In this section, our goal is to integrate the new knowledge into the LoRA module. The new knowledge can be categorized into two parts: one that does not overlap with the original knowledge, and the other that requires updates due to the outdated nature of the original knowledge. The latter is the primary focus, as it may lead to slight changes in the core features. However, these changes should not alter the overall impact of the original knowledge.

To achieve this goal, we constrain the update of parameters while learning new knowledge through an MLP:

$$\begin{aligned} \boldsymbol{O}^{k+1} &= \mathrm{MLP}\left(\boldsymbol{H}^{k+1}\right) \\ &= \delta'\left(\boldsymbol{\omega}^{k+1}\boldsymbol{H}^{k+1} + \boldsymbol{b}^{k+1}\right), \end{aligned} \tag{11}$$

where $\boldsymbol{O}^K$ is the updated matrix $\boldsymbol{A}'$ or $\boldsymbol{B}'$, $\delta'(\cdot)$ is the activation function, $\boldsymbol{\omega}^k \in \boldsymbol{\Omega}$ and $\boldsymbol{b}^k \in \boldsymbol{B}$ are learnable paramters. Defining LoRA-based LLM with pre-trained parameters $\boldsymbol{W}$ and LoRA parameters $\Delta\boldsymbol{W}$ as $\mathcal{A}_{\boldsymbol{W}, \Delta\boldsymbol{W}}(\cdot)$, then the loss function is defined as follows:

$$\begin{aligned} \mathcal{L}oss = \mathrm{MSE}\left(\mathcal{A}_{\boldsymbol{W}, \Delta\boldsymbol{W}'}\left(\boldsymbol{Q}\right),\ \boldsymbol{G}\right) \\ +\lambda_1||\boldsymbol{\theta}_1||_a^a + \lambda_2||\boldsymbol{\theta}_2||_b^b, \end{aligned} \tag{12}$$

where $\Delta\boldsymbol{W}' = \boldsymbol{A}' \times \boldsymbol{B}'^\top$ is parameters of the updated LoRA, $\boldsymbol{G}$ is the ground truth in the new dataset, $\boldsymbol{Q}$ is the input query correspondingly, λ_1 and λ_2 are two regularization hyperparameters, $\boldsymbol{\theta}_1 := \{\boldsymbol{g}, \boldsymbol{C}\}$, $\boldsymbol{\theta}_2 := \{\boldsymbol{\Omega}, \boldsymbol{B}\}$ are the learnable parameters in the model, a and b

are tworegularization orders which satisfies $a \geq b$. By ensuring that the regularization order for the knowledge retention module's parameters is at least as high as that used for the knowledge update module's regularization term, we can minimize the adjustments made to the retention model's parameters. This, in turn, enhances the overall generalization ability of the model.

3.4 Training Procedure

In this subsection, we introduce the training process of DEAL, as shown in Algorithm 1: We begin by locating and obtaining the low-rank matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in LoRA (lines 1–3). Next, we apply wavelet kernel-based neural networks to extract the core features of historical knowledge from $\boldsymbol{A}$ and $\boldsymbol{B}$ (line 5). Then, another neural network is introduced to superimpose features from new knowledge onto these core features, resulting in a newly constructed low-rank matrix $\boldsymbol{A}'$ or $\boldsymbol{B}'$ (lines 6–11). Using this updated low-rank matrix, we recalculate the LoRA parameters to obtain the fine-tuned model. By calculating the loss with regularization terms and performing back-propagation, we ensure that the model can learn new knowledge while preserving historical features in a controlled manner (lines 13–15).

Algorithm 1 The DEAL framework

Input: Pre-trained LoRA-based LLM $\mathcal{A}_{\boldsymbol{W},\Delta\boldsymbol{W}}$, training samples $\mathcal{D}$
Output: Fine-tuned model $\mathcal{A}_{\boldsymbol{W},\Delta\boldsymbol{W}'}$

1: $\Delta\boldsymbol{W}' \leftarrow \Delta\boldsymbol{W}$
2: **for** each batch $(\boldsymbol{Q}, \boldsymbol{G}) \in \mathcal{D}$ **do**
3: Extract $\boldsymbol{A}, \boldsymbol{B}$ from $\Delta\boldsymbol{W}'$
4: **for** each $\boldsymbol{Y} \in \{\boldsymbol{A}, \boldsymbol{B}\}$ **do**
5: $\hat{\boldsymbol{X}} \leftarrow$ Eq. (10), $\boldsymbol{O} \leftarrow$ Eq. (11)
6: **if** $\boldsymbol{Y} == \boldsymbol{A}$ **then**
7: $\boldsymbol{A}' \leftarrow \boldsymbol{O}$
8: **else**
9: $\boldsymbol{B}' \leftarrow \boldsymbol{O}$
10: **end if**
11: **end for**
12: $\Delta\boldsymbol{W}' \leftarrow \boldsymbol{A}' \times \boldsymbol{B}'^{\top}$
13: Compute loss $\mathcal{L}oss$ using Eq. (12)
14: Update parameters via back-propagation
15: **end for**
16: **return** $\mathcal{A}_{\boldsymbol{W},\Delta\boldsymbol{W}'}$

4 Experiments

Datasets. We evaluate **DEAL** on three continual learning (CL) benchmarks in a sequential task setup, where data from previous tasks is unavailable during training on subsequent ones. These benchmarks are designed to evaluate key challenges in continual learning, including: (i) catastrophic forgetting in semantically related tasks, (ii) inefficient data utilization resulting from restricted access to prior samples, and (iii) scalability across long and diverse task sequences.

(i) Same-domain tasks: We use a three-task benchmark consisting of AG News (news classification), DBpedia (entity typing), and Yahoo Answers (question topic prediction). This setup evaluates DEAL's ability to mitigate catastrophic forgetting by retaining transferable knowledge across semantically similar tasks.

(ii) Domain-shift tasks: To introduce domain variability, we augment the benchmark with Amazon Reviews [26] for binary sentiment classification. This domain-shift setting assesses generalization under limited data access and distributional shifts, reflecting practical constraints in real-world continual learning.

(iii) Heterogeneous multi-task learning: We evaluate on the 15-task benchmark proposed by [27], which spans text classification (AG News, DBpedia, Yahoo, Amazon, Yelp), GLUE tasks (MNLI, QQP, RTE, SST-2) [28], SuperGLUE tasks (WiC, CB, COPA, MultiRC, BoolQ) [29], and IMDB [30]. This benchmark tests DEAL's scalability and robustness across heterogeneous tasks and long task sequences. Full details on dataset preprocessing and prompt construction are provided in Appendices B.1 and E.

Baselines. We compare **DEAL** against three LoRA-compatible continual learning baselines:

- **SeqLoRA:** A naive baseline that sequentially updates a single fixed-size LoRA adapter across tasks, without any mechanism to mitigate forgetting.
- **O-LoRA** [31]: A recent method that allocates task-specific adapters to orthogonal subspaces, reducing parameter interference.
- **PerTaskFT:** An oracle baseline in which each task is fine-tuned using a separate LoRA adapter without sharing. Although impractical for deployment, it serves as an upper bound on task-specific retention.

All baselines are evaluated under identical settings, including a fixed adapter architecture, optimization protocol, and tokenization, to ensure fair comparison. Replay-based and non-LoRA methods are excluded to prevent confounding effects from architectural or memory differences. Implementation details are provided in Appendix B.2, and comparisons with broader classes of methods are reported in Appendix G.

Implementation Details. We use two pretrained models: **LLaMA 3.1-8B** [32], a decoder-only model fine-tuned on instruction-following corpora, and **T5-Large-Instruct** [33], an encoder-decoder model adapted for general-purpose instruction following. All experiments are conducted on a single NVIDIA A100 GPU. LoRA-based continual fine-tuning is applied to both models using a fixed adapter rank, dropout, learning rate, and batch size across tasks. Gradients are masked as needed to enforce parameter sparsity. Unless stated otherwise, hyperparameters remain consistent across **DEAL** and all baselines. Full training configurations, hardware details, and random seed settings are provided in Appendix B.2.

Evaluation Metrics. We report **Average Accuracy (AA)**, the mean test accuracy across all tasks after training concludes, and **ROUGE-1 (R-1)**, which measures unigram F1 overlap between generated outputs and ground-truth labels for free-form generation tasks. Formal metric definitions and application contexts are detailed in Appendix D. We evaluate the training and inference efficiency of DEAL in Appendix F.

4.1 Main Results

Table 1 summarizes the continual learning performance of all methods on the three benchmark suites: the 3-task Text Classification (TC), the 4-task Standard CL benchmark, and the 15-task Large-Scale benchmark.

Across all tasks and model backbones, our proposed method, **DEAL**, consistently outperforms both **SeqLoRA** and **O-LoRA**, and achieves performance comparable to the oracle upper bound (PERTASKFT) in terms of average accuracy (AA) and ROUGE-1 F1 (R-1). On the 4-task benchmark with T5-Large, DEAL achieves 78.5% Average Accuracy (AA) and 82.5% ROUGE-1 (R-1), compared to 44.6%/44.6% for SeqLoRA and 71.2%/73.3% for O-LoRA. These improvements stem from DEAL's ability to balance knowledge retention and transfer more effectively. SeqLoRA lacks mechanisms for preserving prior knowledge, leading to severe forgetting. O-LoRA mitigates forgetting via orthogonal subspace constraints but limits beneficial cross-task transfer. In contrast, DEAL integrates shared LoRA modules with dual-branch adapters and stability-aware regularization, enabling robust adaptation while preserving task-specific information.

Table 1: Continual learning performance across three benchmarks.

Method	3-Task (TC)		4-Task (Standard)		15-Task (Large)	
	AA	R-1	AA	R-1	AA	R-1
T5 + SeqLoRA	52.4	52.8	44.6	44.6	42.1	44.0
T5 + O-LoRA	85.2	87.1	71.2	73.3	70.8	80.3
T5 + PerTaskFT	**90.3**	**91.7**	70.0	73.0	**76.5**	**78.2**
T5 + **DEAL(ours)**	87.7	89.3	**78.5**	**82.5**	73.9	79.1
LLaMA + SeqLoRA	54.1	55.9	47.6	54.8	45.2	53.2
LLaMA + O-LoRA	86.4	88.1	75.3	80.8	73.2	77.4
LLaMA + PerTaskFT	**88.2**	**90.0**	77.5	79.4	**77.1**	**82.5**
LLaMA + **DEAL(ours)**	88.9	90.2	**78.9**	**81.3**	74.6	78.9

Note. Bold indicates the statistically significant improvements (*i.e.,* two-sided t-test with $p < 0.05$) over the best baseline.

Notably, DEAL approaches the performance of the oracle baseline, **PerTaskFT**, which fine-tunes each task independently without parameter sharing. On the 3-task benchmark with T5-Large, DEAL achieves 87.7% AA, closely matching the 90.3% attained by PerTaskFT. This near-oracle performance is achieved with significantly lower computational overhead. By leveraging modular adapters and regularized updates, DEAL retains task-discriminative signals while benefiting from shared representations, offering a scalable and efficient alternative to task-isolated fine-tuning.

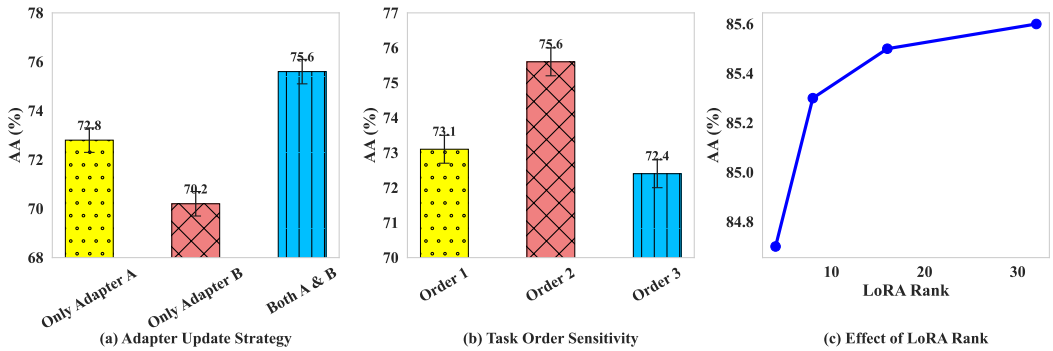

Figure 2: **DEAL** ablations: (a) adapter update strategy, (b) task-order robustness on the 4-task benchmark, (c) LoRA-rank sensitivity on the 3-task benchmark.

The advantages of DEAL become increasingly prominent as task complexity grows. On the 15-task benchmark with LLaMA-3.1-8B, DEAL achieves 74.6% AA, outperforming SeqLoRA by more than 29 percentage points and surpassing O-LoRA as well. In long-horizon continual learning settings, catastrophic forgetting compounds across tasks. DEAL mitigates this degradation through regularization-guided updates and flexible routing, demonstrating strong scalability and robustness to extended task sequences.

5 Ablation Studies

We conduct ablation experiments to assess the contribution of five core components in **DEAL**: (i) adapter update strategy, (ii) task-order robustness, (iii) LoRA rank Figure 2, (iv) the kernel function (Table 3), and (v) the regularization strength (Table 4). Unless otherwise specified, all experiments use the T5-Large backbone with fixed random seeds and data orderings.

Adapter Update Strategy. We evaluate three adapter update strategies: updating only Adapter A, updating only Adapter B, and updating both jointly. Joint updates achieve the highest post-task Average Accuracy (AA) of 75.6%, exceeding single-branch updates by 2.8—5.4 percentage points. Across multiple runs, updating only Adapter A consistently outperforms updating only Adapter B. This result reflects the continual learning setting: Adapter A governs global projection directions that capture generalizable semantic patterns and reasoning structures, whereas Adapter B primarily encodes task-specific features. Joint training of both adapters therefore enables the most effective integration of broad knowledge transfer with task specialization.

Task-Order Robustness. To simulate non-stationary task arrivals, we evaluate three random permutations of the 4-task sequence, following the setup in O-LoRA [31]. This experiment aims to examine how sensitive **DEAL** is to variations in task order—a crucial factor in realistic continual learning, where the order of tasks is typically unpredictable. Across the three permutations, the average accuracy (AA) ranges narrowly from 73.1% to 75.6%, reflecting a fluctuation of less than three percentage points. This low variance demonstrates that **DEAL** maintains stable performance regardless of task order, indicating strong robustness and adaptability to dynamic, non-stationary learning environments. The specific task permutations used are detailed in Table 2.

Table 2: Task sequences used for continual learning evaluations.

Order	Task Sequence
1	DBpedia → Amazon → Yahoo → AG News
2	DBpedia → Amazon → AG News → Yahoo
3	Yahoo → Amazon → AG News → DBpedia

LoRA Rank. Using the 3-task benchmark, we vary the LoRA rank across 4, 8, 16, 32. Accuracy improves significantly from rank 4 to rank 8 (71.5% to 84.3%), then saturates (84.5% at rank 16,

84.6% at rank 32). These results suggest that a compact rank-8 configuration captures most of the task-specific variation while offering strong efficiency in both memory and computation.

Kernel Functions. To further support the selection of the heat kernel, we compared it against two alternatives— $f(x) = xe^{-x}$ and quadratic splines—with results summarized in Table 3. All three kernels achieve comparable accuracy, indicating that the model's representational capacity is relatively insensitive to the specific kernel choice. The heat kernel's primary advantage lies in its computational efficiency: it consistently yields orders-of-magnitude reductions in runtime.

This improvement stems from its inverse-free update rule (Eq. (10)), which avoids expensive matrix inversions while maintaining numerical stability. These results demonstrate that the heat kernel offers a favorable trade-off between simplicity and efficiency, supporting its adoption as the default kernel throughout our experiments.

Table 3: Comparison of different kernel functions.

Kernel Function	Average Accuracy (%)	Training Time (ms/sample)
$f(x) = xe^{-x}$	78.4	1759
Quadratic Splines	78.3	1291
Heat Kernel (Ours)	**78.5**	**56**

Regularization. We conduct a grid search over regularization weights (a, b), where a regulates the retention branch (wavelet-based) and b governs the adaptation branch (MLP-based). The search ranges over $a \in 1, 5, 10, 20$ and $b \in 1, 2, 5$, capturing a variety of retention-to-adaptation penalty combinations. This ablation aims to identify the optimal trade-off between preserving prior knowledge and facilitating new task adaptation.

Table 4: Grid search over asymmetric regularization weights (a, b).

a	b	AA (%)
1	1	74.8
5	1	83.9
10	**2**	**85.5**
10	5	84.1
20	2	82.7

Note. Bold indicates statistically significant improvements (*i.e.,* , two-sided t-test with $p < 0.05$) over the best baseline.

As summarized in Table 4, the best performance is obtained with $(a = 10, b = 2)$, reaching an after-task accuracy (AA) of 85.5%. This configuration outperforms both low-penalty settings (e.g., $(1, 1)$) and heavier regularization schemes (e.g., $(10, 5)$ or $(20, 2)$). These results suggest that appropriately tuning the balance between the retention and adaptation branches is critical: a moderately stronger emphasis on knowledge retention improves stability, while maintaining flexibility in the adaptation pathway supports effective learning of new tasks. Notably, this balance mitigates forgetting without hindering new knowledge acquisition, highlighting the importance of carefully calibrated regularization in continual learning.

6 Related Work

Continual Learning for Large Language Models Continual learning (CL) with large language models (LLMs) presents a fundamental challenge: acquiring new knowledge without catastrophic forgetting, while ensuring efficiency and adaptability. Existing approaches largely fall into three categories: memory-based, regularization-based, and subspace isolation-based methods.

Memory-based approaches, such as experience replay (ER) [34], mitigate forgetting by revisiting buffered past data. However, these methods raise concerns about scalability and data privacy, limiting their practicality in real-world LLM deployments. Regularization-based methods constrain parameter updates to preserve previously learned knowledge. For instance, CLoRA [8] introduces angular regularization between task-specific LoRA adapters. While computationally lightweight, such approaches often under perform on dissimilar tasks due to overly restrictive adaptation dynamics. Subspace-based techniques offer a memory-free alternative by decoupling task representations. O-LoRA [31], for example, updates LoRA parameters within orthogonal subspaces of prior tasks, effectively reducing representational interference. However, it faces two key limitations: (i) orthogonality is enforced only in first-order gradient space, overlooking higher-order interactions; and (ii) all subspace directions are treated uniformly, with no prioritization of semantically meaningful components. The TRACE

benchmark [35] further highlights trade-offs in CL for LLMs: while full-parameter fine-tuning yields high per-task accuracy, it suffers from severe forgetting; in contrast, naive LoRA tuning preserves general capabilities but degrades instruction-following performance.

In contrast to prior work, our method introduces structural regularization and selective routing, aligning task-specific updates with semantically salient directions while preserving cross-task generalization. This design promotes transferability and robustness without relying on external memory or imposing rigid parameter constraints.

Parameter-Efficient Tuning Parameter efficient tuning (PET) approaches, such as adapters [36–38], prompt tuning [39, 40], and LoRA [6, 41], reduce training cost by updating only a small subset of model parameters. Several extensions of LoRA have been proposed to improve adaptability and representational capacity. ReLoRA[42] enables high-rank representations by scheduling low-rank updates across training phases. FLORA[43] employs stochastic resampling to approximate richer adaptation structures with low memory overhead. To support knowledge transfer across tasks, modular PET strategies have also been developed. LoraHub[44] assembles reusable task-specific adapters that can be composed dynamically. MOLE[45] treats multiple LoRA modules as experts and selects among them using a learned gating mechanism.

In contrast to prior work that focuses on isolated improvements in capacity or flexibility, we propose a wavelet regularized continual tuning framework that jointly addresses knowledge retention and adaptive transfer. Our method sustains performance across evolving tasks without relying on data replay, while incurring minimal computational overhead.

7 Conclusion

We present **DEAL**, a continual learning framework that integrates instruction-guided fine-tuning with lightweight adapter updates and structured regularization. Built on a LoRA-style architecture, **DEAL** enables scalable, interference-resistant learning across sequential tasks while maintaining strong performance on diverse benchmarks. Extensive experiments highlight three core advantages:

- **Reduced Forgetting.** DEAL employs regularized low-rank updates to preserve task-relevant subspaces, mitigating catastrophic forgetting without relying on explicit memory buffers.
- **Efficient Adaptation.** Shared LoRA components serve as inductive priors, accelerating convergence and supporting efficient generalization in low-resource scenarios.
- **Scalability.** By introducing only a small number of task-specific parameters, DEAL scales effectively to long task sequences and large LLM backbones such as LLaMA-3.1-8B.

Limitations and Future Directions. While DEAL demonstrates strong performance in diverse continual learning scenarios, it currently assumes a fixed task order and static model capacity. Promising directions include lightweight rehearsal mechanisms, dynamic capacity allocation, and enhancing forward transfer through meta-regularization. Addressing robustness under ambiguous task boundaries remains an open challenge.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (No. 62502404), the Research Impact Fund of the Hong Kong Research Grants Council (No. R1015-23), the Collaborative Research Fund of the Research Grants Council (No. C1043-24GF), the General Research Fund of the Research Grants Council (No. 11218325), and the Institute of Digital Medicine at City University of Hong Kong (No. 9229503). Additional support was provided by Huawei (Huawei Innovation Research Program), Tencent (CCF–Tencent Open Fund; Tencent Rhino-Bird Focused Research Program), Alibaba (CCF–Alimama Tech Kangaroo Fund No. 2024002), Ant Group (CCF–Ant Research Fund), Didi (CCF–Didi Gaia Scholars Research Fund), Kuaishou, and ByteDance.

References

[1] Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409, 2022.

[2] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023.

[3] Yuqian Fu, Runze Wang, Yanwei Fu, Danda Pani Paudel, Xuanjing Huang, and Luc Van Gool. Objectrelator: Enabling cross-view object relation understanding in ego-centric and exo-centric videos. *ICCV*, 2025.

[4] Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. *Advances in Neural Information Processing Systems*, 37:26701–26727, 2024.

[5] Zichuan Fu, Xiangyang Li, Chuhan Wu, Yichao Wang, Kuicai Dong, Xiangyu Zhao, Mengchen Zhao, Huifeng Guo, and Ruiming Tang. A unified framework for multi-domain ctr prediction via large language models. *ACM Transactions on Information Systems*, 43(5):1–33, 2025.

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[7] Maolin Wang, Jun Chu, Sicong Xie, Xiaoling Zang, Yao Zhao, Wenliang Zhong, and Xiangyu Zhao. Put teacher in student's shoes: Cross-distillation for ultra-compact model compression framework. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 4975–4985, 2025.

[8] James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *arXiv preprint arXiv:2304.06027*, 2023.

[9] D Chen. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

[10] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

[11] Xuanwen Ding, Jie Zhou, Liang Dou, Qin Chen, Yuanbin Wu, Chengcai Chen, and Liang He. Boosting large language models with continual learning for aspect-based sentiment analysis. *arXiv preprint arXiv:2405.05496*, 2024.

[12] Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Cheng-Zhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37:9565–9584, 2024.

[13] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

[14] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022.

[15] Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18564–18572, 2024.

[16] Shuaiyi Li, Yang Deng, Deng Cai, Hongyuan Lu, Liang Chen, and Wai Lam. Consecutive batch model editing with hook layers. *arXiv preprint arXiv:2403.05330*, 2024.

[17] Yujie Feng, Liming Zhan, Zexin Lu, Yongxin Xu, Xu Chu, Yasha Wang, Jiannong Cao, Philip S Yu, and Xiao-Ming Wu. Geoedit: Geometric knowledge editing for large language models. *arXiv preprint arXiv:2502.19953*, 2025.

[18] Zichuan Fu, Wentao Song, Yejing Wang, Xian Wu, Yefeng Zheng, Yingying Zhang, Derong Xu, Xuetao Wei, Tong Xu, and Xiangyu Zhao. Sliding window attention training for efficient large language models. *arXiv preprint arXiv:2502.18845*, 2025.

[19] Minsoo Kim, Sihwa Lee, Wonyong Sung, and Jungwook Choi. RA-LoRA: Rank-adaptive parameter-efficient fine-tuning for accurate 2-bit quantized large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15773–15786, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.933. URL `https://aclanthology.org/2024.findings-acl.933/`.

[20] Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Jiahuan Pei. Melora: mini-ensemble low-rank adapters for parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.17263*, 2024.

[21] Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *arXiv preprint arXiv:2409.05976*, 2024.

[22] Yimin Tian, Bolin Zhang, Zhiying Tu, and Dianhui Chu. Adapters selector: Cross-domains and multi-tasks LoRA modules integration usage method. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 593–605, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL `https://aclanthology.org/2025.coling-main.40/`.

[23] Zijian Zhang, Shuchang Liu, Jiaao Yu, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Ziru Liu, Qidong Liu, Hongwei Zhao, Lantao Hu, et al. M3oe: Multi-domain multi-task mixture-of experts recommendation framework. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 893–902, 2024.

[24] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[25] Kai Liao and Yan Xu. A robust load frequency control scheme for power systems based on second-order sliding mode and extended disturbance observer. *IEEE Transactions on industrial informatics*, 14(7):3076–3086, 2017.

[26] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

[27] Akari Asai, Sneha Kudugunta, Xinyan Velocity Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. Buffet: Benchmarking large language models for few-shot cross-lingual transfer. *arXiv preprint arXiv:2305.14857*, 2023.

[28] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[29] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[30] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://aclanthology.org/P11-1015/`.

[31] 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. *arXiv preprint arXiv:2310.14152*, 2023.

[32] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[33] 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.

[34] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. In *Neural Information Processing Systems*, 2018. URL `https://api.semanticscholar.org/CorpusID:53860287`.

[35] Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, et al. Trace: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*, 2023.

[36] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

[37] Maolin Wang, Tianshuo Wei, Sheng Zhang, Ruocheng Guo, Wanyu Wang, Shanshan Ye, Lixin Zou, Xuetao Wei, and Xiangyu Zhao. Dance: Resource-efficient neural architecture search with data-aware and continuous adaptation. *arXiv preprint arXiv:2507.04671*, 2025.

[38] Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1104–1114, 2024.

[39] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[40] Wenxin Luo, Weirui Wang, Xiaopeng Li, Weibo Zhou, Pengyue Jia, and Xiangyu Zhao. Tapo: Task-referenced adaptation for prompt optimization. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[41] Maolin Wang, Xiangyu Zhao, Ruocheng Guo, and Junhui Wang. Metalora: Tensor-enhanced adaptive low-rank fine-tuning. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pages 4680–4684. IEEE, 2025.

[42] Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. Relora: High-rank training through low-rank updates. *arXiv preprint arXiv:2307.05695*, 2023.

[43] Chi-Chih Chang, Yuan-Yao Sung, Shixing Yu, Ning-Chi Huang, Diana Marculescu, and Kai-Chiang Wu. Flora: Fine-grained low-rank architecture search for vision transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2482–2491, 2024.

[44] Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*, 2023.

[45] Xun Wu, Shaohan Huang, and Furu Wei. Mixture of lora experts. *arXiv preprint arXiv:2404.13628*, 2024.

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

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

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

[49] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149, 2022.

[50] Chengwei Qin and Shafiq Joty. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. *arXiv preprint arXiv:2110.07298*, 2021.

[51] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*, 2023.

[52] Fuli Qiao and Mehrdad Mahdavi. Learn more, but bother less: parameter efficient continual learning. *Advances in Neural Information Processing Systems*, 37:97476–97498, 2024.

Appendix

A Supplementary Algorithmic Analysis

A.1 Notation

Let $\boldsymbol{X} \in \mathbb{R}^{n_x \times r}$ be a matrix of rank r_x. Its singular value decomposition (SVD) is:

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{U}_x \boldsymbol{\Sigma}_x \boldsymbol{V}_x^\top \\ &= [\boldsymbol{U}_{x1}\ \boldsymbol{U}_{x2}] \begin{bmatrix} \boldsymbol{\Sigma}_{x1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{x1}^\top \\ \boldsymbol{V}_{x2}^\top \end{bmatrix}. \end{aligned} \tag{13}$$

Here, $\boldsymbol{U}_{x1} \in \mathbb{R}^{n_x \times r_x}$ and $\boldsymbol{V}_{x1} \in \mathbb{R}^{r \times r_x}$ correspond to the principal subspace, while $\boldsymbol{U}_{x2}$ and $\boldsymbol{V}_{x2}$ span the orthogonal complement.

A.2 Proof of Theorem 1

Consider a perturbation $\boldsymbol{D} \in \mathbb{R}^{n_x \times r}$, and define:

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{D}. \tag{14}$$

We decompose $\boldsymbol{D}$ via projection onto the column spaces of $\boldsymbol{V}_{x1}$ and $\boldsymbol{V}_{x2}$:

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{X} + \boldsymbol{D}\left(\boldsymbol{V}_{x1}\boldsymbol{V}_{x1}^\top + \boldsymbol{V}_{x2}\boldsymbol{V}_{x2}^\top\right) \\ &= \left(\boldsymbol{X}\boldsymbol{V}_{x1} + \boldsymbol{D}\boldsymbol{V}_{x1}\right)\boldsymbol{V}_{x1}^\top + \left(\boldsymbol{D}\boldsymbol{V}_{x2}\right)\boldsymbol{V}_{x2}^\top. \end{aligned} \tag{15}$$

Denoting SVDs of each term:

$$\boldsymbol{X}\boldsymbol{V}_{x1} + \boldsymbol{D}\boldsymbol{V}_{x1} = \boldsymbol{P}_1 \boldsymbol{S}_1 \boldsymbol{Q}_1^\top \tag{16}$$

$$\boldsymbol{D}\boldsymbol{V}_{x2} = \boldsymbol{P}_2 \boldsymbol{S}_2 \boldsymbol{Q}_2^\top, \tag{17}$$

with $\boldsymbol{P}_1^\top \boldsymbol{P}_2 = 0$, we can express $\boldsymbol{Y}$ as:

$$\boldsymbol{Y} = [\boldsymbol{P}_1\ \boldsymbol{P}_2] \begin{bmatrix} \boldsymbol{S}_1 & 0 \\ 0 & \boldsymbol{S}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}_1^\top \boldsymbol{V}_{x1}^\top \\ \boldsymbol{Q}_2^\top \boldsymbol{V}_{x2}^\top \end{bmatrix}. \tag{18}$$

This constitutes the SVD of $\boldsymbol{Y}$, and the basis $\boldsymbol{P}_1$ generally differs from $\boldsymbol{U}_{x1}$ due to perturbation. This completes the proof.

B Experimental Setup

B.1 Datasets

Table 5 summarizes all datasets used across the continual learning benchmarks. These datasets were also employed in O-LoRA [31], where each is framed as a classification task using a unified instruction-based text-to-text format.

Table 5: Overview of datasets used in experiments.

#	Dataset	Source	Task Type	Evaluation Metric
1	AG News	CL Benchmark	Topic Classification	Accuracy
2	DBpedia	CL Benchmark	Entity Typing	Accuracy
3	Yahoo	CL Benchmark	Topic Classification	Accuracy
4	Amazon	CL Benchmark	Sentiment Analysis	Accuracy
5	MNLI	GLUE	Natural Language Inference	Accuracy
6	QQP	GLUE	Paraphrase Detection	Accuracy
7	RTE	GLUE	Natural Language Inference	Accuracy
8	SST-2	GLUE	Sentiment Analysis	Accuracy
9	WiC	SuperGLUE	Word Sense Disambiguation	Accuracy
10	CB	SuperGLUE	Natural Language Inference	Accuracy
11	COPA	SuperGLUE	Causal Reasoning	Accuracy
12	BoolQ	SuperGLUE	Boolean QA	Accuracy
13	MultiRC	SuperGLUE	Multi-hop QA	Accuracy
14	IMDB	External	Sentiment Analysis	Accuracy

B.2 Implementation Details

We implement **DEAL** using the Hugging Face Transformers library and perform training with FP16 mixed precision on a single NVIDIA A100 GPU. Unless otherwise stated, we adopt a consistent experimental setup across both LLaMA-3.1 and T5-large backbones, tailored for low-resource, instruction-driven continual learning. All models are trained with adapter-based fine-tuning using LoRA, where we set the rank $r = 32$ for LLaMA and $r = 16$ for T5, selected based on backbone capacity. Optimization is performed using AdamW with a constant learning rate scheduler. Regularization is enforced via ℓ_p-norm constraints on adapter weights and MLP modules, with details provided in Table 6.

Table 6: Hyperparameter settings for **DEAL** on LLaMA-3.1 and T5-large.

Hyperparameter	**LLaMA-3.1**	**T5-large**
LoRA Rank r	32	16
Learning Rate	1e-5	1e-5
Batch Size	8	4
Gradient Accum. Steps	4	2
Epochs	1	1
Max Source Length	512	512
Max Target Length	50	50
Generation Max Length	50	50
Warmup Steps	0	0
Dropout	–	–
Optimizer	AdamW	AdamW
Scheduler Type	constant	constant
Regularization λ_1	0.01	0.01
Regularization λ_2	0.001	0.001
$\|\theta\|_5$ Norm Reg.	✓	✓
$\|\text{MLP}\|_2$ Norm Reg.	✓	✓
LoRA Modules	q_proj, v_proj	q, v
Task Type	CausalLM	Seq2SeqLM

We set $\lambda_1 = 0.01$ to constrain the ℓ_5-norm of adapter parameters θ, and $\lambda_2 = 0.001$ to regularize the ℓ_2-norm of task-specific MLP weights, when applicable. All experiments are conducted for one epoch over instruction-based task mixtures, without early stopping or checkpointing unless explicitly mentioned.

C Case Study Examples

Case 1: DBpedia Classification Task

Prompt:
You are a smart AI evaluator. Given an input paragraph and three model outputs *(Base Model, Adapter after DBpedia, Adapter after DBpedia→Amazon)*, judge which output(s) correctly classify the paragraph according to the true label.

Question:
What is the topic of the following paragraph? Choose one from the options below.
["Company", "Educational Institution", "Artist", "Athlete", "Office Holder", "Mean of Transportation", "Building", "Natural Place", "Village", "Animal", "Plant", "Album", "Film", "Written Work"]

Input (DBpedia sample)

Label: Office Holder

Text: Raimundas Palaitis (born 23 October 1957) is a Lithuanian politician. He was Minister of the Interior from 2008 to 2012.

Base Model: Building

Adapter after DBpedia: Office Holder

Adapter after DBpedia→Amazon: Office Holder

Expected Answer:
The base model incorrectly predicted "Building," which is unrelated to the paragraph. In contrast, both adapters correctly predicted "Office Holder," indicating that they effectively learned the DBpedia task and that the second adapter retained this knowledge even after subsequent training on the Amazon task

Case 2: Amazon Sentiment Task

Prompt:
Given an input review and three model outputs, decide whether the sentiment is classified correctly and briefly explain why.

Question:
What is the sentiment of the following review? Choose one from
["very negative", "negative", "neutral", "positive", "very positive"]

Input (Amazon sample)

Label: very negative

Text: I don't understand how they can advertise that this humidifier can work up to 12 hours. It runs out of water so fast. I have to get up in the middle of the night and refill it. I would say it works well for about 2 hours. So, while it is inexpensive, you get what you pay for.

Base Model: neutral

Adapter after DBpedia: neutral

Adapter after DBpedia→Amazon: very negative

Expected Answer:
Both the base model and the first adapter misclassified the review as "neutral". Only the adapter further fine-tuned on Amazon data predicted the correct label, "very negative," demonstrating its ability to acquire new task knowledge that the earlier models failed to capture.

Figure 3: Two case studies demonstrating continual learning and knowledge retention across classification tasks.

D Evaluation Metrics

We adopt the following standard metrics for continual learning evaluation:

- **Average Accuracy (AA)**: Measures the average test accuracy across all tasks after the final task is learned:
$$\mathrm{AA} = \frac{1}{T}\sum_{i=1}^{T} a_{i,T},$$
where $a_{i,T}$ is the test accuracy on task i after training on task T.
- **ROUGE-1 (R-1)**: Used for generative label decoding, computed as the unigram F_1 between model output and reference:
$$\mathrm{R}{-}1 = \frac{2 \cdot P \cdot R}{P + R}, \quad P = \frac{|y \cap y^\star|}{|y|}, \quad R = \frac{|y \cap y^\star|}{|y^\star|}.$$

E Instruction Prompts

We adopt the task-specific instruction prompts introduced in O-LoRA [31], as summarized in Table 7.

Table 7: Instruction prompts provided to the model for each task.

Task	Prompt
NLI	*What is the logical relationship between "sentence 1" and "sentence 2"? Choose one from the options.*
QQP	*Do "sentence 1" and "sentence 2" express the same meaning? Choose one from the options.*
SC	*What is the sentiment of the following passage? Choose one from the options.*
TC	*What is the topic of the following passage? Choose one from the options.*
BoolQA	*According to the passage, is the statement true or false? Choose one from the options.*
MultiRC	*Based on the passage and question, is the candidate's answer correct? Choose one from the options.*
WiC	*Given a word and two sentences, is the word used with the same sense in both? Choose one from the options.*

F Training and Inference Efficiency

Training Efficiency. We measure the training throughput and GPU memory usage on the DBpedia dataset with the T5-large backbone (Table 8).

Table 8: Training efficiency on DBpedia with T5-large.

Method	Training Throughput (samples/sec)	GPU Mem. Train (GB)
LoRA	31.62	20.41
DEAL	17.88	22.93

Inference Efficiency. We evaluate inference latency and GPU memory usage under the same setting (Table 9).

Table 9: Inference efficiency on DBpedia with T5-large.

Method	Inference Latency (ms/sample)	GPU Mem. Infer (GB)
LoRA	71.89	3.15
DEAL	73.32	3.16

Discussion. As shown in Tables 8 and 9, DEAL incurs a $\sim$43% drop in training throughput and a small increase in training GPU memory, while inference-time performance remains nearly identical to LoRA in both latency and memory usage. Since the wavelet module is only active during training and enables consistent performance gains (see Table 1), this overhead represents a worthwhile trade-off for better generalization and robustness.

G Comparison with Additional Continual Learning Methods

To provide a comprehensive evaluation, we expand our comparisons to include a wide range of strong non-LoRA continual learning (CL) baselines. Table 10 reports the average accuracy (AA) on a standard CL benchmark using the T5-large backbone.

Our method, DEAL, achieves the highest average accuracy while maintaining parameter efficiency. It outperforms both rehearsal-based approaches (Replay) and regularization-based techniques (EWC, LwF). Additionally, DEAL surpasses prompt-based strategies (L2P, ProgPrompt) and recent competitive baselines (LFPT5, LB-CL), highlighting its effectiveness as a general-purpose solution for continual learning.

Table 10: Comparison with Additional Continual Learning Methods.

Method	**Average Accuracy (AA)**
SeqSVD	63.3
Replay	52.0
EWC	45.3
LwF	52.9
L2P	60.5
LFPT5	71.2
ProgPrompt	76.0
LB-CL	76.5
DEAL (Ours)	**78.5**

- **SeqSVD**: Learns a fixed-size SVD parameter space across sequential tasks without regularization or replay.
- **Replay**: Rehearses past samples to prevent forgetting by storing and replaying them during training [46].
- **EWC**: Mitigates forgetting by penalizing changes to important parameters estimated via Fisher information [47].
- **LwF**: Preserves knowledge of past tasks through knowledge distillation without storing old data [48].
- **L2P**: Introduces learnable prompts for continual learning to guide pre-trained models effectively [49].
- **LFPT5**: Proposes a unified framework for lifelong few-shot learning based on prompt tuning of T5 [50].
- **ProgPrompt**: Designs progressive prompts to adapt large language models for continual learning [51].
- **LB-CL**: Achieves parameter-efficient continual learning by learning more but disturbing less [52].

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

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 the "Main Results" paragraph in Section 1 accurately reflect the key contributions and scope of 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: A summary of the key limitations of this work is provided in Section 7.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Appendix A lists all the assumptions used and provides the complete proof.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: A comprehensive description of the experimental setup and hyperparameters is included in Appendix B.

   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: We include detailed instructions for running the experiments, and the code is publicly released as open source.

   Guidelines:

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

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The dataset splits, hyperparameter configurations, and detailed experimental setup are thoroughly documented in Appendix B.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We evaluate performance in Section 4.1 using three continual learning benchmarks across different models.

   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: Section 4 provides a clear report of the computational resources used in our experiments.

Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The paper follows 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: The broader impacts of our approach are discussed in detail in Section 6.

    Guidelines:

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

11. **Safeguards**

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

Answer: [NA]

Justification: No datasets or models are released in this work.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The dataset we use is completely open source. In addition, the method mentioned in the paper is brand new and is also the innovation of this article.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: We release our code as open source and include detailed instructions to facilitate its use.

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 usage of LLMs is illustrated in detail in the Method and Experiments section.

    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.