Estimating and Orthogonalizing Unknown Pre-training Gradients for Continual Fine-tuning of Large Language Models

Bing Wang[1,2], **Changchun Li**[1,2], **Xin-Qiang Cai**[3], **Lin Yuanbo Wu**[4], **Ximing Li**[1,2,3]*, **Gang Niu**[3] , **Masashi Sugiyama**[3,5]
[1] College of Computer Science and Technology, Jilin University
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering, MoE, Jilin University
[3] RIKEN Center for Advanced Intelligence Project [4] University of Warwick
[5] Graduate School of Frontier Sciences, University of Tokyo
{wangbing1416,changchunli93,liximing86}@gmail.com

Abstract

Continual fine-tuning is essential for large language models (LLMs) to dynamically adapt to real-world environments, yet it inevitably suffers from catastrophic forgetting, particularly the performance degradation of previous tasks and LLMs' general-purpose knowledge. Although existing methods, such as orthogonal gradient projection, mitigate the forgetting across various fine-tuning tasks, they fundamentally fail to preserve pre-training LLMs' inherent general-purpose knowledge because the original *data and gradients of off-the-shelf pre-training LLMs required by these methods are strictly unknown and highly diverse*. To bridge this critical gap, we propose EOUPCT, a novel framework designed to *Estimate and Orthogonalize Unknown Pre-training gradients for Continual LLM fine-Tuning*. Specifically, EOUPCT estimates pre-training gradients by dynamically generating pseudo data that is most susceptible to forgetting for new tasks through a learnable soft prompt equipped with Gumbel-Softmax relaxation. Furthermore, we formulate a multi-objective optimization problem and introduce a first-order efficient Pareto optimizer that jointly optimizes LLM parameters and the soft prompt, rigorously enforcing orthogonality between new task updates and the estimated pre-training gradients. Extensive experiments across multiple LLMs demonstrate that EOUPCT effectively preserves both task-specific proficiency and inherent general-purpose knowledge, successfully mitigating the catastrophic forgetting.

1 Introduction

Recently, the community has witnessed the rapid development of *large language models* (LLMs) in solving a variety of downstream tasks, *e.g.,* scientific Q&A [26] and mathematical reasoning [4, 35]. However, in dynamic real-world environments, the ceaseless influx of knowledge across emerging tasks makes it difficult for off-the-shelf pre-training LLMs to timely adapt to new information. To bridge this gap, *continual LLM fine-tuning* on a stream of new tasks presents a pivotal solution [3, 29].

Despite the continuous acquisition of new task-specific knowledge, the primary challenge in continual LLM fine-tuning remains *catastrophic forgetting*, that is, training on new tasks leads to a sustained performance decline on previously learned tasks and a degradation of the pre-training LLM's inherent general-purpose knowledge [20, 18, 23, 28]. To address this issue, a typical solution in the continual learning community is the *orthogonal gradient optimization* approach [31, 29, 17]. Generally, this

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

method involves storing the gradients of previously learned tasks and projecting the training gradients of the new task onto the orthogonal subspace spanned by these stored gradients, thereby mitigating interference between parameter updates for different tasks. For instance, *orthogonal low-rank adaptation* (OLoRA) [31] allocates a distinct set of LoRA parameters [7] to each new task and enforces the orthogonality across the parameters of different task modules.

Although these orthogonalization methods are effective in mitigating catastrophic forgetting across various downstream tasks, *they fail to account for the preservation of the pre-training LLMs' inherent general-purpose knowledge*, *e.g.,* commonsense knowledge and logic reasoning. In practice, maintaining pre-trained general-purpose knowledge is also computationally or architecturally infeasible for existing methods. Specifically, as *the training data and gradients of off-the-shelf pre-training LLMs are typically unknown*, employing techniques such as gradient orthogonalization is rendered unrealistic. Furthermore, *the tasks involved in pre-training LLM training are highly diverse* [16], even if recent techniques could estimate the underlying pre-training gradients by generating pseudo training data [2, 10, 21], identifying which gradients are most beneficial remains a critical challenge.

To address these challenges, we propose a novel method to *Estimate and Orthogonalize Unknown Pre-training gradients for Continual LLM fine-Tuning*, namely EOUPCT. Specifically, EOUPCT involves two key steps. **First**, we estimate pre-training gradients by generating pseudo data [2, 38] through a learnable soft prompt [13] utilized as the prefix to directly prompt the pre-training LLM. Since the direct auto-regressive generation of discrete tokens of the pseudo data is non-differentiable w.r.t. the soft prompt, we apply a Gumbel-Softmax relaxation [9] to derive continuous token representations. **Second**, we formulate a multi-objective optimization problem to jointly optimize the soft prompt and the task-specific LLM parameters. Specifically, to optimize the LLM parameters, we not only perform continual fine-tuning on the new tasks but also enforce orthogonality between task-specific gradients, as well as between new task gradients and the estimated pre-training gradients. To optimize the soft prompt, we aim to generate pseudo data that is most susceptible to forgetting for each new task, while ensuring orthogonality among the estimated pre-training gradients to represent diverse pre-training tasks. To solve this complex multi-objective optimization problem, we further propose a first-order efficient Pareto optimizer, avoiding the computational burden of solving second-order optimization problems.

To evaluate the performance of EOUPCT, we sample 15 natural language tasks from the *SuperNI* dataset [32]. We evaluate task-specific proficiency using the *SuperNI* test subset and assess general-purpose knowledge via the *MMLU* benchmark [6]. Generally, EOUPCT achieves a 3.5% improvement in average accuracy and a 3.7% reduction in forgetting rate against SOTA results across 6 different pre-training LLMs, demonstrating that our approach consistently mitigates the catastrophic forgetting. *Our source code is released in* https://github.com/wangbing1416/EoupCT.

In summary, our contributions can be summarized as the following three-folds:

- We observe that while orthogonal gradient projection approaches mitigate interference between fine-tuning tasks, they struggle to preserve the pre-training LLMs' general-purpose knowledge, primarily due to the unknown and diverse nature of the pre-training gradients.
- To address unknown and diverse pre-training gradients during continual fine-tuning, we propose EOUPCT, which estimates and orthogonalizes these pre-training gradients.
- Extensive experiments demonstrate that EOUPCT significantly preserves both task-specific proficiency and general-purpose knowledge, thereby mitigating the catastrophic forgetting.

2 Related Work

In this section, we briefly summarize the related literature about recent continual LLM fine-tuning methods. Generally, existing research primarily mitigates the catastrophic forgetting issue during the continual LLM fine-tuning through several series of methods: *orthogonal gradient projection*, which restricts parameter updates to lie within non-interfering parameter directions [31, 37, 17, 29]. *Data rehearsal* involves continuously storing training samples from historical tasks and training them concurrently with the new task to mitigate their catastrophic forgetting [24, 33, 8, 5]. For example, self-synthesized rehearsal (SSR) [8, 30] utilizes a pre-training LLM to synthesize queries and employs a state-of-the-art LLM to generate corresponding responses as historical task data, which is then utilized for retraining during new task iterations. However, this approach remains unable to access

pre-training data for the purpose of experience replay. *Parameter isolation*, *e.g.,* low-rank adaptation-based mixture-of-experts (LoRAMoE) [3, 19, 40, 36], which allocates independent parameters for each task to isolate the influence between them [15, 12, 27, 22, 39]. While this approach decouples pre-trained weights from new task parameters, it necessitates maintaining a number of historical LLMs for routing during inference. Furthermore, achieving a balance between general-purpose knowledge and task-specific expertise remains challenging; as a unified model, even with parameter isolation, the integration of new tasks inevitably induces forgetting of general-purpose knowledge.

Among these, our work focuses on orthogonal gradient projection, a promising direction that ensures optimization directions do not interfere with general-purpose knowledge in pre-training weights. However, existing methods face two critical limitations. **First**, many current approaches do not operate directly on the gradients; instead, they often apply constraints to parameter weights, *e.g.,* OLoRA [31] or hidden feature subspaces, *e.g.,* InfLoRA [14], which may not optimally represent the dynamic optimization landscape. **Second**, and more importantly, existing methods only consider the gradients of known sequential tasks within the training stream [29, 17, 34], which typically ignore the pre-training gradients because the pre-training data is unknown. This neglect leads to the erosion of the LLM's general-purpose knowledge during continual adaptation. To address these issues, we propose EOUPCT that estimates the missing pre-training gradients, thereby enabling a more comprehensive orthogonal projection that protects both general-purpose and task-specific knowledge.

3 Our Proposed EOUPCT

In this section, we introduce our proposed method EOUPCT that orthogonally optimize LLMs.

Task definition of continual LLM fine-tuning. Let $\mathcal{F}_{\boldsymbol{\theta}^0}$ denote a pre-training LLM, which is pre-trained on a set of datasets $\mathcal{T}_0$ encompassing general-purpose knowledge. The goal of continual fine-tuning is to adapt this LLM to a sequence of downstream tasks. Specifically, we consider a sequence of datasets $[\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T]$ representing T different tasks, where each task $\mathcal{T}_t = \{(\mathbf{x}_i^t, \mathbf{y}_i^t)\}_{i=1}^{|\mathcal{T}_t|}$ consists of input prompts $\mathbf{x}_i^t$ and target outputs $\mathbf{y}_i^t$. The LLM, parameterized by $\boldsymbol{\theta}$ and initialized as $\boldsymbol{\theta}^0$, is then fine-tuned across this sequence by minimizing the empirical loss. A critical issue in this process is the mitigation of catastrophic forgetting. This necessitates the preservation of both the general-purpose knowledge inherent in $\mathcal{F}_{\boldsymbol{\theta}^0}$, represented by $\mathcal{T}_0$, and the task-specific knowledge acquired from preceding tasks $\mathcal{T}_{1:t-1}$ during the fine-tuning of the current task dataset $\mathcal{T}_t$.

Formulation of orthogonal gradient projection methods. To fundamentally mitigate the interference between the current task $\mathcal{T}_t$ and previous tasks $\mathcal{T}_{1:t-1}$, orthogonal gradient projection restricts parameter updates to the orthogonal subspace spanned by previous inputs. Let $\boldsymbol{\theta}$ denote the trainable LLM parameters. During the fine-tuning on $\mathcal{T}_t$, we first compute the standard empirical gradient $\mathbf{g}_t = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{T}_t)$. Instead of applying this gradient directly, we project it using an orthogonal projection matrix $\mathbf{H}_{\perp}(\mathcal{T}_{1:t-1})$, which projects $\mathbf{g}_t$ onto the null space of the gradients $\mathbf{g}_{1:t-1}$ from all previous data $\mathcal{T}_{1:t-1}$. The parameter update rule is thus formulated as $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \mathbf{g}_t \mathbf{H}_{\perp}(\mathcal{T}_{1:t-1})$, where η is the learning rate. By mathematically ensuring that the projected gradient yields zero activation for any historical input $\mathbf{x} \in \mathcal{T}_{1:t-1}$, this method theoretically guarantees that the new parameter updates act as a null operator on previously acquired knowledge, therefore preventing catastrophic forgetting.

3.1 Method Overview

Although the orthogonal gradient projection methods prevent interference between the gradients of the current task $\mathcal{T}_t$ and previous tasks $\mathcal{T}_{1:t-1}$, they fail to account for orthogonality relative to the gradients of the pre-training task $\mathcal{T}_0$, as the latter remain *unknown*. Therefore, it is challenging to enforce orthogonality between the gradients of new tasks and those of pre-training tasks, which leads to a degradation of general-purpose knowledge. To address this, the basic idea of EOUPCT is to *estimate pre-training gradients from the pre-training LLM $\mathcal{F}_{\boldsymbol{\theta}^0}$, and orthogonally optimize the gradients of tasks $\mathcal{T}_{1:T}$ against the pre-training gradients*. Specifically, EOUPCT is an end-to-end training strategy that consists of two pivotal modules: ***pre-training gradient estimation***, which directly prompts the pre-training LLM to generate pseudo data and estimates pre-training gradients, and ***Pareto orthogonal optimization***, which formulates a multi-objective optimization problem to ensure that the gradients of new tasks remain orthogonal to the estimated pre-training gradients.

Formally, given an off-the-shelf pre-training LLM $\mathcal{F}_{\boldsymbol{\theta}^0}$, and a trainable LLM parameterized by $\boldsymbol{\theta}$, *pre-training gradient estimation* introduces a continuous L-length soft prompt $\mathbf{P}_t$ for the current

task $\mathcal{T}_t$ into the pre-training LLM $\mathcal{F}_{\boldsymbol{\theta}^0}$ to generate pseudo data of the general-purpose knowledge distribution $\mathcal{T}_0$. Since discrete auto-regressive generation is non-differentiable during orthogonal optimization, we employ a Gumbel-Softmax relaxation to generate a continuous, fully differentiable surrogate sequence $\mathbf{S}(\mathbf{P}_t)$. This continuous generation maintains an unbroken computational graph, which is mathematically essential for the subsequent backpropagation optimization of the prompt $\mathbf{P}_t$. Accordingly, we define a knowledge distillation loss $\mathcal{L}_{\text{pre}}(\mathbf{S}(\mathbf{P}_t))$ between the pre-training and trained LLMs on the pseudo data, and estimate the pre-training gradient $\mathbf{g}_{\text{pre}}(\mathbf{P}_t) = \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{pre}}$.

Given the estimated gradients, *Pareto orthogonal optimization* formulates the continual LLM fine-tuning process as an end-to-end multi-objective optimization problem. The objective is to jointly optimize the soft prompt $\mathbf{P}_t$ and the LLM parameters $\boldsymbol{\theta}$ to balance new task plasticity and general-purpose knowledge stability. Specifically, to achieve this goal, we present the following ***Constraints***. Given a new task $\mathcal{T}_t$, **[C1] orthogonality**: typically, addressing the forgetting requires ensuring gradient orthogonality not only between the new tasks during the optimization of $\boldsymbol{\theta}$, but also between the new tasks and the estimated pre-training gradients. Meanwhile, during the training of $\mathbf{P}_t$, orthogonality must be kept between the current and previous estimated pre-training gradients, to ensure that the pseudo data generated by different soft prompts represent distinct pre-training tasks. **[C2] Vulnerability**: we aim for the generated pseudo data sequence $\mathbf{S}(\mathbf{P}_t)$ to effectively expose the general-purpose knowledge that is most susceptible to being forgotten during the training of the current new task, *i.e.,* $\mathbf{max}_{\mathbf{P}_t}\mathcal{L}_{\text{new}}(\mathcal{T}_t)$. **[C3] Plasticity**: given the generated pseudo data sequence $\mathbf{S}(\mathbf{P}_t)$ and new task data $\mathcal{T}_t$, we perform supervised fine-tuning on $\boldsymbol{\theta}$ using these datasets, *i.e.,* $\mathbf{min}_{\boldsymbol{\theta}}\mathcal{L}_{\text{new}}(\mathcal{T}_t), \mathcal{L}_{\text{pre}}(\mathbf{S}(\mathbf{P}_t))$.

Finally, our formulation in Appendix B.1 demonstrates that the optimization of the aforementioned objective necessitates the computation of a second-order Hessian matrix, rendering it computationally intractable. Therefore, we propose a first-order Pareto optimizer to efficiently address this multi-objective optimization problem. For clarity, *the overall framework of* EOUPCT *is depicted in Fig. 1, and the complete algorithm of* EOUPCT *is described in Appendix C*. In the following subsections, we provide a detailed introduction to pre-training gradient estimation, Pareto optimization objectives, and the efficient Pareto optimizer, respectively.

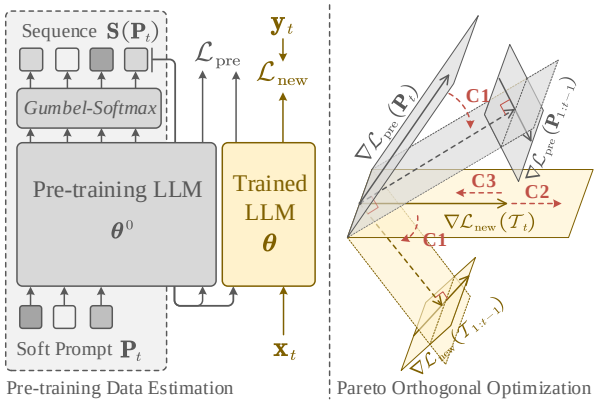

Figure 1: Framework of EOUPCT with pre-training gradient estimation and Pareto orthogonal optimization.

3.2 Pre-training Gradient Estimation

As motivated in Sec. 1, approximating the orthogonal complement of the general-purpose knowledge subspace requires access to the pre-training data distribution $\mathcal{T}_0$, which is unknown during continual LLM fine-tuning. To address this bottleneck, we propose a *differentiable generation* mechanism that treats the off-the-shelf pre-training LLM $\mathcal{F}_{\boldsymbol{\theta}^0}$ itself as an implicit generative model of $\mathcal{T}_0$ [2, 38].

Specifically, for the current fine-tuning task $\mathcal{T}_t$, we introduce a learnable L-length soft prompt $\mathbf{P}_t \in \mathbb{R}^{L\times D}$, where D indicates the hidden size, that is prepended to $\mathcal{F}_{\boldsymbol{\theta}^0}$ to steer it toward generating sequences of the general-purpose knowledge distribution. However, standard auto-regressive decoding relies on discrete token selection, *e.g.,* top-k sampling, which is non-differentiable and would sever the gradient flow $\nabla_{\mathbf{P}_t}$ essential for the optimization of the prompt $\mathbf{P}_t$. To maintain a fully differentiable computational graph, we adopt the Gumbel-Softmax relaxation [9] for differentiable decoding. Let $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{|\mathcal{V}|\times D}$ denote the token embedding matrix, where $\mathcal{V}$ is the token vocabulary. We auto-regressively generate a sequence of M soft token embeddings $\mathbf{S}(\mathbf{P}_t) = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M]$. At generation step $m \in \{1, \ldots, M\}$, the pre-training LLM $\mathcal{F}_{\boldsymbol{\theta}^0}$ processes the sequence $[\mathbf{P}_t, \mathbf{e}_1, \ldots, \mathbf{e}_{m-1}]$ to produce the next-token logits $\mathbf{z}_m \in \mathbb{R}^{|\mathcal{V}|}$. Instead of selecting a discrete token from $\mathbf{z}_m$, we compute a continuous probability vector via the Gumbel-Softmax relaxation as

$$\pi_{m,i} = \frac{\exp\left((z_{m,i} + \varepsilon_{m,i})/\tau\right)}{\sum_{j=1}^{|\mathcal{V}|}\exp\left((z_{m,j} + \varepsilon_{m,j})/\tau\right)}, \tag{1}$$

where $\tau > 0$ is the temperature parameter (as $\tau \to 0$, $\boldsymbol{\pi}_m$ converges to a one-hot vector), and $\varepsilon_{m,i} = -\log(-\log(u_{m,i}))$ with $u_{m,i} \sim \text{Uniform}(0,1)$ are Gumbel noise samples. The corresponding soft token embedding is obtained by computing the expected embedding under $\boldsymbol{\pi}_m$:

$$\mathbf{e}_m = \boldsymbol{\pi}_m \mathbf{W}_{\text{emb}} = \sum_{i=1}^{|\mathcal{V}|} \pi_{m,i} (\mathbf{W}_{\text{emb}})_i \in \mathbb{R}^D, \tag{2}$$

which is appended to the prefix for the next decoding step. After M iterations, this process yields the pseudo data $\mathbf{S}(\mathbf{P}_t)$. Accordingly, every operation from the initial soft prompt $\mathbf{P}_t$ to the final sequence $\mathbf{S}(\mathbf{P}_t)$ is strictly continuous and differentiable, making $\mathbf{P}_t$ optimizable. Accordingly, we define a knowledge distillation loss based on the *mean squared error* between the latent representations of the pre-training and trained LLMs $\boldsymbol{\theta}^0$ and $\boldsymbol{\theta}$ on the pseudo data:

$$\mathcal{L}_{\text{pre}}(\boldsymbol{\theta}; \mathbf{S}(\mathbf{P}_t)) = \ell_{\text{MSE}} \left(\mathcal{F}_{\boldsymbol{\theta}^0}(\mathbf{S}(\mathbf{P}_t)),\ \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{S}(\mathbf{P}_t)) \right), \tag{3}$$

with the corresponding estimated pre-training gradient $\mathbf{g}_{\text{pre}}(\mathbf{P}_t) = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{pre}}$.

3.3 Multi-Objective Formulation of Pareto Orthogonal Optimization

Given the pseudo data sequence $\mathbf{S}(\mathbf{P}_t)$ in Sec. 3.2 and the new task data $\mathcal{T}_t$, Pareto orthogonal optimization formalizes the three conditions **[C1-C3]** outlined in Sec. 3.1 into a unified *multi-objective optimization* problem to jointly optimize the soft prompt $\mathbf{P}_t$ and the LLM parameter $\boldsymbol{\theta}$. Specifically, we formulate the following objectives.

⋆ **Plasticity objective [C3].** To learn the new task $\mathcal{T}_t$, we minimize the standard supervised fine-tuning loss over the task dataset. Formally, the plasticity objective is defined as the negative log-likelihood:

$$\mathcal{L}_{\text{new}}(\boldsymbol{\theta}; \mathcal{T}_t) = -\frac{1}{|\mathcal{T}_t|} \sum_{i=1}^{|\mathcal{T}_t|} \sum_{k=1}^{|\mathbf{y}_i|} \log P_{\boldsymbol{\theta}} \left(y_{i,k}^t \mid \mathbf{x}_i^t, \mathbf{y}_{i,<k}^t \right), \quad \left(\mathbf{x}_i^t, \mathbf{y}_i^t \right) \in \mathcal{T}_t, \tag{4}$$

where $P_{\boldsymbol{\theta}}\left(y_{i,k}^t \mid \mathbf{x}_i^t, \mathbf{y}_{i,<k}^t \right)$ denotes the probability of generating the k-th token $y_{i,k}^t$ conditioned on the input $\mathbf{x}_i^t$. This yields the task-specific gradient $\mathbf{g}_{\text{new}} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{new}}$.

⋆ **Stability objective [C3].** We minimize $\mathcal{L}_{\text{pre}}(\boldsymbol{\theta}; \mathbf{S}(\mathbf{P}_t))$ in Eq. (3) to keep the general-purpose knowledge stability. To ensure that the pseudo data in Sec. 3.2 is informative and that the parameter update is orthogonal across previous gradients, we further impose the following three constraints.

⋆ **Vulnerability [C2].** The soft prompt $\mathbf{P}_t$ must dynamically expose general-purpose knowledge that is most susceptible to being forgotten during the current task $\mathcal{T}_t$. Instead of arbitrary estimation, we optimize $\mathbf{P}_t$ by maximizing $\mathcal{L}_{\text{pre}}$ on a *virtually updated* $\boldsymbol{\theta}_{\text{virt}} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{new}}$. This virtual step simulates the destructive impact of the new task's gradient, forcing $\mathbf{P}_t$ to generate pseudo data that specifically exposes the general-purpose knowledge boundaries most damaged by the current task.

⋆ **Pre-training & new gradients orthogonality [C1].** To resolve the forgetting, the new task gradient must not interfere with the currently estimated pre-training gradient $\mathbf{g}_{\text{pre}}(\mathbf{P}_t)$. This is inherently formulated as: finding an update direction that minimizes the new task loss without increasing the loss on the optimal $\mathbf{S}(\mathbf{P}_t^*)$, effectively enforcing orthogonality when gradient conflicts arise.

⋆ **Pre-training & pre-training gradients orthogonality [C1].** To ensure diverse coverage of the general-purpose knowledge subspace, current pseudo data must not redundantly overlap with previous generations. We penalize the squared cosine similarity between the current protection gradient $\mathbf{g}_{\text{pre}}(\mathbf{P}_t)$ and historical pre-training gradients. This explicitly drives $\langle \mathbf{g}_{\text{pre}}(\mathbf{P}_t), \mathbf{g}_{\text{pre}}(\mathbf{P}_{1:t-1}) \rangle \to 0$, protecting distinct regions of the pre-training distribution.

⋆ **New & new gradients orthogonality [C1].** To prevent forgetting of previous downstream tasks, the actual parameter update $\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t$ must reside entirely within the orthogonal null space of $\mathcal{M}_{\text{new}} = \{\mathbf{g}_{\text{new}}^1, \mathbf{g}_{\text{new}}^2, \ldots, \mathbf{g}_{\text{new}}^{t-1}\}$. This mathematically guarantees $\langle \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t, \mathbf{g}_{\text{new}}^i \rangle = 0$ for all historical gradients $\mathbf{g}_{\text{new}}^i \in \mathcal{M}_{\text{new}}$, ensuring absolute zero interference.

Integrating these objectives and constraints, the entire continual LLM fine-tuning process is cast as the following unified multi-objective optimization formulation:

$$\min_{\boldsymbol{\theta}} \Big(\underbrace{\mathcal{L}_{\text{new}}(\boldsymbol{\theta};\, \mathcal{T}_t)}_{\text{New task plasticity}},\ \underbrace{\mathcal{L}_{\text{pre}}\big(\boldsymbol{\theta};\, \mathbf{S}(\mathbf{P}_t^*)\big)}_{\text{General-purpose stability}} \Big) \tag{5}$$

$$\text{s.t. } \mathbf{P}_t^* = \arg\max_{\mathbf{P}_t} \Big[\underbrace{\mathcal{L}_{\text{pre}}\big(\boldsymbol{\theta}_{\text{virt}};\, \mathbf{S}(\mathbf{P}_t)\big)}_{\text{Vulnerability}} - \underbrace{\lambda \sum_{i=1}^{t-1} \cos^2\big(\mathbf{g}_{\text{pre}}(\mathbf{P}_t),\, \mathbf{g}_{\text{pre}}(\mathbf{P}_i)\big)}_{\text{Pre-training \& pre-training gradients orthogonality}} \Big], \tag{6}$$

$$\underbrace{\big\langle \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t,\, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{pre}}\big(\boldsymbol{\theta};\, \mathbf{S}(\mathbf{P}_t^*)\big) \big\rangle \le 0}_{\text{Pre-training \& new gradients orthogonality}},\ \underbrace{\big\langle \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t,\, \mathbf{g}_{\text{new}}^{1:t-1} \big\rangle = 0}_{\text{New \& new gradients orthogonality}}. \tag{7}$$

This formulation encodes the goals of continual fine-tuning into a tractable mathematical structure. We next describe an efficient first-order optimizer for this second-order optimization problem.

3.4 Efficient Pareto Orthogonal Optimizer

Directly solving the multi-objective optimization problem defined in Eqs. (5), (6), and (7) is computationally intractable. This stems from the natural trade-off between vulnerability and stability; furthermore, explicitly coupling adversarial prompt optimization with new-task gradients would necessitate the computation of second-order Hessian matrices. Therefore, we resolve this by decoupling the optimizer into four efficient first-order steps, corresponding directly to our defined constraints. The complete algorithm of this efficient optimizer is shown in Appendix C.

Step 1: To satisfy both the vulnerability and the pre-training & pre-training gradients orthogonality in Eq. (6), we optimize the soft prompt $\mathbf{P}_t$. A naïve approach, maximizing $\mathcal{L}_{\text{pre}}$ at the current weights $\boldsymbol{\theta}_t$, only finds knowledge that the current model already struggles to preserve, completely ignoring the destructive effect of new tasks. The correct objective is to maximize the gradient conflict, *i.e.,* $\max_{\mathbf{P}_t} \langle -\mathbf{g}_{\text{new}},\, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{pre}} \rangle$; however, computing its gradient w.r.t. $\mathbf{P}_t$ requires an expensive Hessian matrix. We circumvent this via the *virtual step approximation*: by a first-order Taylor expansion around the current weights $\boldsymbol{\theta}_t$, taking a virtual step $\boldsymbol{\theta}_{\text{virt}} = \boldsymbol{\theta}_t - \alpha\, \mathbf{g}_{\text{new}}$ changes the loss as

$$\mathcal{L}_{\text{pre}}(\boldsymbol{\theta}_{\text{virt}}) \approx \mathcal{L}_{\text{pre}}(\boldsymbol{\theta}_t) - \alpha \big\langle \mathbf{g}_{\text{new}},\, \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{pre}}(\boldsymbol{\theta}_t) \big\rangle. \tag{8}$$

Hence, maximizing $\mathcal{L}_{\text{pre}}(\boldsymbol{\theta}_{\text{virt}};\, \mathbf{S}(\mathbf{P}_t))$ over $\mathbf{P}_t$ is provably equivalent to maximizing the gradient conflict. Simultaneously, to enforce the orthogonality among estimated pre-training distributions, we subtract the historical independence penalty $\lambda \sum_{i=1}^{t-1} \cos^2\big(\mathbf{g}_{\text{pre}}(\mathbf{P}_t),\, \mathbf{g}_{\text{pre}}(\mathbf{P}_i)\big)$ from this objective during the backpropagation of $\mathbf{P}_t$. Once the optimal prompt $\mathbf{P}_t^*$ is obtained via this joint optimization, the true protection gradient is computed as $\mathbf{g}_{\text{pre}}^* = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{pre}}\big(\boldsymbol{\theta}_t;\, \mathbf{S}(\mathbf{P}_t^*)\big)$.

Step 2: With both $\mathbf{g}_{\text{new}}$ and $\mathbf{g}_{\text{pre}}^*$ in hand, we strictly enforce the orthogonality with new gradient in Eq. (7) by constructing the null-space projection matrix of the historical new-task gradients $\mathcal{M}_{\text{new}}$:

$$\boldsymbol{\Pi}_{\text{new}} = \mathbf{I} - \mathbf{M}_{\text{new}} \big(\mathbf{M}_{\text{new}}^\top \mathbf{M}_{\text{new}} \big)^{-1} \mathbf{M}_{\text{new}}^\top, \tag{9}$$

where $\mathbf{M}_{\text{new}}$ denotes the matrix by concatenating historical task gradients $\mathcal{M}_{\text{new}}$ as column vectors, and $\mathbf{I}$ is the identity matrix. Then, we project both raw gradients into this historically safe subspace:

$$\tilde{\mathbf{g}}_{\text{new}} = \boldsymbol{\Pi}_{\text{new}}\, \mathbf{g}_{\text{new}}, \quad \tilde{\mathbf{g}}_{\text{pre}} = \boldsymbol{\Pi}_{\text{new}}\, \mathbf{g}_{\text{pre}}^*. \tag{10}$$

This guarantees that any subsequent parameter update built from these projected gradients will have exactly zero inner product with any previously learned task.

Step 3: To satisfy the pre-training & new gradients orthogonality in Eq. (7), we must ensure the new task update does not increase the loss on the generated pseudo data. We compute the alignment score, if $\langle \tilde{\mathbf{g}}_{\text{new}},\, \tilde{\mathbf{g}}_{\text{pre}} \rangle \ge 0$, the two gradients are naturally aligned. If $\langle \tilde{\mathbf{g}}_{\text{new}},\, \tilde{\mathbf{g}}_{\text{pre}} \rangle < 0$, updating along $\tilde{\mathbf{g}}_{\text{new}}$ would damage the adversarially identified general knowledge. We resolve this by applying a Pareto-optimal projection that removes the conflicting component of $\tilde{\mathbf{g}}_{\text{new}}$ along the direction of $\tilde{\mathbf{g}}_{\text{pre}}$:

$$\tilde{\mathbf{g}}_{\text{new}}^* = \tilde{\mathbf{g}}_{\text{new}} - \frac{\langle \tilde{\mathbf{g}}_{\text{new}},\, \tilde{\mathbf{g}}_{\text{pre}} \rangle}{\|\tilde{\mathbf{g}}_{\text{pre}}\|^2}\, \tilde{\mathbf{g}}_{\text{pre}}. \tag{11}$$

A direct calculation confirms $\langle \tilde{\mathbf{g}}_{\text{new}}^*,\, \tilde{\mathbf{g}}_{\text{pre}} \rangle = 0$, mathematically guaranteeing that the modified task gradient avoids any interference with the currently estimated pre-training gradient.

Table 1: Results of EOUPCT under two task types: task-specific knowledge by *SuperNI* and general-purpose knowledge by *MMLU* (ROU. ↑, Acc. ↑, Fgt. ↓). The bold results represent the best scores.

	Method	***SuperNI***						***MMLU*** (average three task orders)					
		Order 1		*Order 2*		*Order 3*		*STEM*		*Humanity*		*Other*	
		ROU.	Fgt.	ROU.	Fgt.	ROU.	Fgt.	Acc.	Fgt.	Acc.	Fgt.	Acc.	Fgt.
Qwen3-8B	+ LoRA	50.1	7.7	48.2	9.8	49.9	8.5	62.6	3.5	71.3	7.6	72.5	4.4
	+ LoRAMoE	50.8	6.3	49.4	8.7	48.9	8.4	61.6	4.5	71.3	7.5	72.0	4.9
	+ OLoRA	51.2	6.7	47.4	9.4	50.1	7.6	63.3	2.9	72.0	6.9	72.9	4.0
	+ GainLoRA	50.0	6.6	49.7	8.0	49.2	8.1	62.7	3.4	72.7	6.2	72.4	4.4
	+ CLoRA	50.9	7.2	49.6	9.1	49.2	7.6	63.3	2.8	72.5	6.4	72.2	4.7
	+ EOUPCT	**52.9**	**4.4**	**52.1**	**6.0**	**51.5**	**5.7**	**64.7**	**1.4**	**75.9**	**3.0**	**75.5**	**1.4**
Qwen3-4B	+ LoRA	45.1	13.7	40.8	16.7	48.6	9.2	56.4	4.5	64.0	8.6	64.2	5.7
	+ LoRAMoE	48.6	8.8	43.2	13.9	48.1	8.8	57.4	3.6	65.1	7.4	65.7	4.1
	+ OLoRA	48.9	9.0	46.0	10.6	47.9	9.6	58.1	2.9	65.4	7.1	65.9	3.9
	+ GainLoRA	48.4	9.0	44.9	12.1	47.8	9.6	55.9	5.0	64.8	7.8	65.3	4.6
	+ CLoRA	47.8	9.9	45.4	11.3	47.2	10.0	57.7	3.3	64.1	8.4	64.3	5.6
	+ EOUPCT	**50.7**	**6.0**	**50.3**	**6.1**	**50.5**	**6.2**	**60.2**	**0.7**	**67.4**	**5.2**	**68.0**	**1.9**
Llama3-8B	+ LoRA	35.5	22.0	33.9	23.7	40.8	16.8	39.0	7.0	58.0	11.1	63.1	9.3
	+ LoRAMoE	42.9	14.7	36.7	20.5	37.5	20.2	40.3	5.7	59.1	10.0	64.3	8.1
	+ OLoRA	44.6	12.8	43.7	12.8	45.6	12.2	43.6	2.4	62.2	7.0	65.7	6.7
	+ GainLoRA	40.2	14.5	36.8	18.6	37.4	17.4	40.9	5.1	59.0	10.0	65.3	7.1
	+ CLoRA	41.6	14.7	37.2	18.2	41.5	13.5	42.1	3.9	62.0	7.0	64.9	7.5
	+ EOUPCT	**51.4**	**6.5**	**49.0**	**7.8**	**50.2**	**7.4**	**45.0**	**1.0**	**64.5**	**4.6**	**69.6**	**2.8**
Llama3-3B	+ LoRA	37.6	17.7	39.3	14.4	43.7	10.7	38.9	1.7	56.0	6.1	59.5	4.2
	+ LoRAMoE	40.5	14.2	40.1	16.2	39.4	16.1	39.4	1.2	56.0	6.1	60.8	3.7
	+ OLoRA	41.6	13.5	36.4	20.0	36.7	18.3	39.2	1.4	55.6	6.5	60.0	2.9
	+ GainLoRA	38.0	16.3	38.3	18.1	43.0	11.6	38.4	2.2	55.8	6.3	59.3	4.3
	+ CLoRA	38.7	15.3	39.2	15.6	39.9	13.2	39.4	1.2	55.4	6.7	59.8	3.9
	+ EOUPCT	**44.0**	**10.5**	**45.1**	**9.2**	**46.5**	**8.2**	**41.4**	**-0.8**	**57.2**	**4.9**	**61.9**	**1.7**
Gemma2-9B	+ LoRA	41.2	17.7	41.3	16.8	40.0	19.1	41.7	11.5	60.7	13.9	64.3	12.8
	+ LoRAMoE	48.8	10.2	36.6	23.0	45.5	13.7	46.8	7.4	65.7	8.9	68.7	8.4
	+ OLoRA	50.0	8.8	44.9	14.4	40.7	17.0	42.9	10.3	61.8	12.8	67.7	9.5
	+ GainLoRA	43.3	12.9	40.7	17.0	42.2	14.4	44.3	8.9	65.6	9.0	67.9	9.2
	+ CLoRA	42.8	13.3	40.8	15.5	42.5	12.7	45.1	8.1	64.8	9.8	64.8	12.3
	+ EOUPCT	**53.4**	**4.8**	**53.9**	**5.5**	**54.3**	**4.5**	**52.3**	**0.9**	**71.8**	**2.7**	**71.7**	**5.4**
Gemma2-2B	+ LoRA	34.7	20.8	35.5	20.4	28.5	27.3	31.0	8.3	45.1	11.7	52.9	14.0
	+ LoRAMoE	36.0	20.9	34.4	22.2	38.3	17.2	33.0	6.3	48.9	7.9	59.6	7.3
	+ OLoRA	39.7	16.5	36.2	19.6	40.3	15.5	33.1	6.2	47.9	8.9	58.0	8.9
	+ GainLoRA	41.7	14.4	41.4	14.9	43.4	12.4	34.8	4.5	50.1	6.7	59.4	7.5
	+ CLoRA	37.5	17.7	39.2	16.8	38.8	17.1	34.9	4.4	47.4	9.3	58.1	8.8
	+ EOUPCT	**46.7**	**10.3**	**48.1**	**8.4**	**50.9**	**4.6**	**38.5**	**0.8**	**54.4**	**2.5**	**63.0**	**3.9**

Step 4: With the interference-free gradients established, the LLM parameters are updated as

$$\boldsymbol{\theta}_{\tau+1} = \boldsymbol{\theta}_{\tau} - \eta\big(\tilde{\mathbf{g}}^{*}_{\text{new}} + \tilde{\mathbf{g}}_{\text{pre}}\big), \tag{12}$$

where $\eta > 0$ represents the learning rate. Because $\tilde{\mathbf{g}}^{*}_{\text{new}}$ and $\tilde{\mathbf{g}}_{\text{pre}}$ reside in the null space of $\mathcal{M}_{\text{new}}$ (by Step 2) and do not conflict with each other (by Step 3), this update direction $\boldsymbol{\theta}_{\tau+1} - \boldsymbol{\theta}_{\tau}$ simultaneously satisfies Eq. (7), efficiently breaking the catastrophic forgetting.

4 Experimental Evaluation

In this section, we present the experimental performance of our EOUPCT method.

Experimental settings. To evaluate the performance of our method, we conduct experiments using *SuperNI* [32], a large-scale multi-task instruction tuning dataset. Following Zhao et al. [40], we select 15 tasks organized into 3 different sequential orderings for continual LLM fine-tuning. Upon the completion of each task's fine-tuning phase, we evaluate the LLM on the corresponding *SuperNI* test subsets to assess task-specific performance. Furthermore, we utilize 9 tasks across 3 categories from the *MMLU* dataset [6] to benchmark the preservation of the LLM's general-purpose knowledge. In our implementation, we utilize 6 pre-training LLMs across 3 distinct families, including Qwen3, Llama3,

and Gemma2, employing *low-rank adaptation* (LoRA) [7] with a rank of $r = 4$ for parameter-efficient fine-tuning. We compare our approach against 5 baselines, encompassing a variety of continual fine-tuning strategies. Detailed dataset statistics, descriptions of the LLMs and baseline methods, implementation specifics, and formal evaluation metrics are provided in Appendix D.

4.1 Main Results

Table 1 presents the overall performance of EOUPCT compared with a diverse set of continual LLM fine-tuning baselines across multiple LLM families and scales. We conduct fine-tuning across three distinct task sequences in Table 6 generated using 3 different random seeds, reporting the task-specific performance on *SuperNI* for each individual ordering. Meanwhile, the general-purpose knowledge performance on *MMLU* is reported as the average across these three task sequences.

Generally, we can observe that EOUPCT consistently achieves superior performance in both task-specific learning and general-purpose knowledge preservation. On *SuperNI*, our method yields the highest average ROUGE scores under all task orders while significantly reducing forgetting. For instance, on Qwen3-4B, EOUPCT improves the ROUGE of OLoRA from 48.9 to 50.7 under Order 1, and simultaneously reduces the forgetting from 9.0 to 6.0. Similar improvements are consistently observed across different task sequences, indicating that the proposed method is robust to the order of incoming tasks. More importantly, EOUPCT demonstrates a substantial advantage in mitigating catastrophic forgetting. Across all model families, the forgetting values are consistently lower than those of competing methods. This verifies that explicitly enforcing orthogonality with estimated pre-training gradients effectively prevents destructive interference during continual updates.

Meanwhile, EOUPCT also shows remarkable improvements in preserving general-purpose knowledge on *MMLU*. Compared to existing methods, which often suffer from severe degradation in general-purpose knowledge, EOUPCT achieves both higher accuracy and lower forgetting across all categories. For example, on Gemma2-9B, EOUPCT improves the STEM accuracy of OLoRA from 42.9 to 52.3 while reducing forgetting from 10.3 to 0.9. Notably, these gains are consistently observed across different domains, including Humanity and Other, suggesting that the estimated pre-training gradients effectively capture diverse general knowledge.

4.2 Sensitivity Analysis

We further investigate the sensitivity of EOUPCT to two key hyper-parameters: the soft prompt length L and the surrogate sequence length M. Specifically, we conduct experiments on Qwen3-4B and Llama3-3B, and report the performance in terms of ROUGE on the *SuperNI* benchmark under Order 1. Both L and M are selected from the set $\{4, 8, 16, 32\}$ to explore their impact on model performance.

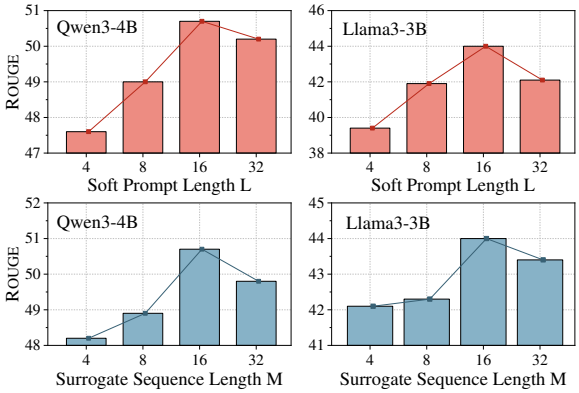

Figure 2: Sensitivity analysis of soft prompt length L and surrogate sequence length M.

The results show a consistent trend across both LLMs. When L and M are small, *e.g.,* 4 or 8, the model exhibits relatively weaker performance, suggesting that limited prompt capacity and short generated sequences are insufficient to adequately cover diverse pre-training knowledge. As L and M increase, the performance improves steadily, indicating that a richer soft prompt and longer surrogate sequences enable more expressive and representative reconstruction of the underlying pre-training distribution. However, when L and M become excessively large, *e.g.,* 32, the performance does not continue to improve and may even slightly degrade.

4.3 Ablation Experiments

We further conduct ablation studies to analyze the contribution of each component in EOUPCT, as shown in Table 2. Removing any component leads to a consistent degradation in both task-specific performance and general knowledge preservation, highlighting the necessity of the full design. In

Table 2: Ablation experiments of different constraints in EOUPCT.

Ablation	Qwen3-4B				Llama3-3B				Gemma2-2B			
	SuperNI		*MMLU*		*SuperNI*		*MMLU*		*SuperNI*		*MMLU*	
	ROU.	Fgt.	Acc.	Fgt.	ROU.	Fgt.	Acc.	Fgt.	ROU.	Fgt.	Acc.	Fgt.
EOUPCT	**50.7**	**6.0**	**60.2**	**0.7**	**44.0**	**10.5**	**41.4**	**-0.8**	**46.7**	**10.3**	**38.5**	**0.8**
w/o C2	47.6	10.3	59.0	2.0	42.3	13.0	40.4	0.2	44.8	10.1	36.9	2.3
w/o P&PO	47.3	9.2	57.8	3.1	42.1	12.2	40.3	0.3	43.5	13.0	36.9	2.4
w/o N&PO	47.4	8.7	57.9	3.0	41.9	12.2	40.0	0.6	44.2	13.3	36.6	2.6
w/o N&NO	46.9	10.3	56.9	4.1	41.5	11.7	39.6	1.0	43.2	13.5	35.7	3.6

particular, removing the *vulnerability objective,* i.e., *w/o C2* results in a noticeable drop in ROUGE and a sharp increase in forgetting. For example, on Qwen3-4B, the ROUGE decreases from 50.7 to 47.6 while the forgetting increases from 6.0 to 10.3, indicating that identifying the most vulnerable pre-training knowledge is crucial for effective protection.

Similarly, *removing the orthogonality constraints between pre-training gradients,* i.e., *w/o P&PO* leads to further performance degradation. On Llama3-3B, the ROUGE drops from 44.0 to 42.1 and forgetting increases from 10.5 to 12.2, demonstrating that enforcing orthogonality with estimated pre-training gradients plays a key role in preserving general-purpose knowledge. When *removing the orthogonality between new-task and pre-training gradients,* i.e., *w/o N&PO*, the LLM suffers from increased interference, resulting in higher forgetting across all settings. Moreover, *removing the orthogonality among new-task gradients,* i.e., *w/o N&NO* leads to the most severe degradation, as it directly breaks the protection against inter-task interference. This confirms that maintaining orthogonality across sequential tasks is essential for stable continual learning. Across all ablations, we also observe consistent declines on *MMLU* performance, indicating that each component contributes to preserving general-purpose knowledge.

4.4 Forgetting Dynamics

To further analyze the forgetting behavior during continual fine-tuning, we track the evolution of general-purpose knowledge by evaluating the accuracy on three *MMLU* categories as the *SuperNI* tasks are sequentially learned. We compare EOUPCT with the recent method CLoRA, focusing on how the performance changes throughout the training process.

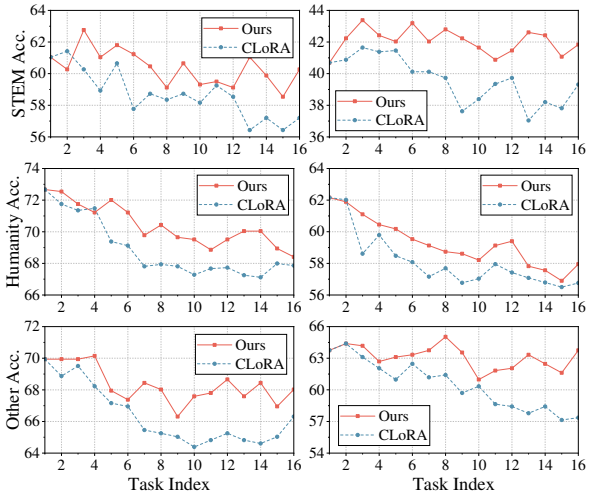

Figure 3: Forgetting dynamics of Qwen3-4B (left) and Llama3-3B (right) across *MMLU*.

The results reveal a clear advantage of EOUPCT in terms of both the rate and extent of forgetting. As the number of learned tasks increases, CLoRA exhibits a steady decline in *MMLU* accuracy, indicating progressive degradation of general-purpose knowledge. In contrast, EOUPCT demonstrates a much slower decrease, maintaining consistently higher accuracy across all stages of training. This confirms that our method effectively mitigates the accumulation of forgetting over time. More interestingly, we observe that EOUPCT not only preserves knowledge but can even improve performance in certain stages, where the *MMLU* accuracy temporarily surpasses that of the initial pre-trained LLM before any fine-tuning. This phenomenon suggests that the generated pseudo data and the associated orthogonal optimization not only prevent knowledge erosion but may also reinforce and refine useful representations within the LLM.

5 Conclusion

In this paper, we address the challenge of catastrophic forgetting in the continual LLM fine-tuning, where existing methods fail to preserve inherent general-purpose knowledge due to the unknown and highly diverse nature of original pre-training data and gradients. To overcome this fundamental barrier, we propose EOUPCT, a novel framework that dynamically estimates and orthogonalizes these unknown pre-training representations. Specifically, our approach leverages a learnable soft prompt equipped with a Gumbel-Softmax relaxation to generate the pseudo pre-training data most susceptible to forgetting. Furthermore, we formulate a multi-objective optimization problem solved by an efficient first-order Pareto optimizer, which jointly optimizes the soft prompt and task-specific LLM parameters to rigorously enforce gradient orthogonality without the burden of second-order computations. Extensive experiments across multiple LLMs demonstrate that EOUPCT effectively maintains task-specific proficiency while preserving the models' broad general-purpose capabilities, successfully offering a robust solution to the catastrophic forgetting in continual LLM fine-tuning.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No.62276113, No.62372150). X-Q C was supported by JSPS KAKENHI Grant Number JP26K21323. MS was supported by JST ASPIRE Grant Number JPMJAP2405.

References

[1] AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[2] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, pages 2633–2650. USENIX Association, 2021.

[3] S. Dou, E. Zhou, Y. Liu, S. Gao, W. Shen, L. Xiong, Y. Zhou, X. Wang, Z. Xi, X. Fan, S. Pu, J. Zhu, R. Zheng, T. Gui, Q. Zhang, and X. Huang. Loramoe: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Annual Meeting of the Association for Computational Linguistics*, pages 1932–1945, 2024.

[4] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081): 633–638, 2025.

[5] Y. He, W. Zhang, X. Huang, P. Zhang, L. Meng, X. Zhou, K. Zeng, and X. Cai. Don't half-listen: Capturing key-part information in continual instruction tuning. In *Annual Meeting of the Association for Computational Linguistics*, pages 23649–23668, 2025.

[6] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[8] J. Huang, L. Cui, A. Wang, C. Yang, X. Liao, L. Song, J. Yao, and J. Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Annual Meeting of the Association for Computational Linguistics*, pages 1416–1428, 2024.

[9] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[10] A. M. Kassem, O. Mahmoud, N. Mireshghallah, H. Kim, Y. Tsvetkov, Y. Choi, S. Saad, and S. Rana. ALPACA AGAINST VICUNA: using llms to uncover memorization of llms. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, pages 8296–8321, 2025.

[11] G. Khromov and S. P. Singh. Some fundamental aspects about lipschitz continuity of neural networks. In *International Conference on Learning Representations*, 2024.

[12] Y. Leng and D. Xiong. Towards understanding multi-task learning (generalization) of llms via detecting and exploring task-specific neurons. In *International Conference on Computational Linguistics*, pages 2969–2987, 2025.

[13] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

[14] Y. Liang and W. Li. Inflora: Interference-free low-rank adaptation for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23638–23647, 2024.

[15] Y.-S. Liang, J. Chen, and W.-J. Li. Gated integration of low-rank adaptation for continual learning of large language models. In *Advances in Neural Information Processing Systems*, 2025.

[16] C. Liao, R. Xie, X. Sun, H. Sun, and Z. Kang. Exploring forgetting in large language model pre-training. In *Annual Meeting of the Association for Computational Linguistics*, pages 2112–2127, 2025.

[17] Y. Lu, B. Qian, C. Yuan, H. Jiang, and X. Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Annual Meeting of the Association for Computational Linguistics*, pages 19165–19181, 2025.

[18] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *CoRR*, abs/2308.08747, 2023.

[19] Y. Ma, Z. Liang, H. Dai, B. Chen, D. Gao, Z. Ran, Z. Wang, L. Jin, W. Jiang, G. Zhang, X. Cai, and L. Yang. Modula: Mixture of domain-specific and universal lora for multi-task learning. In *Conference on Empirical Methods in Natural Language Processing*, pages 2758–2770, 2024.

[20] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation-Advances in Research and Theory*, 24(C):109–165, 1989.

[21] M. Nasr, J. Rando, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, F. Tramèr, and K. Lee. Scalable extraction of training data from aligned, production language models. In *International Conference on Learning Representations*, 2025.

[22] F. Qiao and M. Mahdavi. Merge before forget: A single lora continual learning via continual merging. In *International Conference on Learning Representations*, 2026.

[23] H. Shi, Z. Xu, H. Wang, W. Qin, W. Wang, Y. Wang, Z. Wang, S. Ebrahimi, and H. Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):120:1–120:42, 2026.

[24] F. Sun, C. Ho, and H. Lee. LAMOL: language modeling for lifelong language learning. In *International Conference on Learning Representations*, 2020.

[25] G. Team. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118, 2024.

[26] Q. Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

[27] B. Wang, X. Li, M. Ye, C. Li, B. Fu, J. Qu, and L. Y. Wu. Remember past, anticipate future: Learning continual multimodal misinformation detectors. In *ACM International Conference on Multimedia*, pages 5490–5498, 2025.

[28] B. Wang, X. Li, C. Li, J. Chi, G. Niu, and M. Sugiyama. Decomposing the basic abilities of large language models: Mitigating cross-task interference in multi-task instruct-tuning. In *International Conference on Machine Learning*, pages 127780–127804, 2026.

[29] C. Wang, Y. Lyu, Z. Sun, and L. Jing. Continual gradient low-rank projection fine-tuning for llms. In *Annual Meeting of the Association for Computational Linguistics*, pages 14815–14829, 2025.

[30] J. Wang, L. Ding, S. Wang, H. Li, Y. Luo, H. Zhao, H. Hu, and B. Du. Self-evolving pseudo-rehearsal for catastrophic forgetting with task similarity in llms. In *Advances in Neural Information Processing Systems*, 2025.

[31] X. Wang, T. Chen, Q. Ge, H. Xia, R. Bao, R. Zheng, Q. Zhang, T. Gui, and X. Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 10658–10671, 2023.

[32] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Naik, A. Ashok, A. S. Dhanasekaran, A. Arunkumar, D. Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, 2022.

[33] Y. Wang, Y. Liu, C. Shi, H. Li, C. Chen, H. Lu, and Y. Yang. Inscl: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 663–677, 2024.

[34] Y. Xiong and X. Xie. Oplora: Orthogonal projection lora prevents catastrophic forgetting during parameter-efficient fine-tuning. In *AAAI Conference on Artificial Intelligence*, pages 34088–34096, 2026.

[35] S. Yan, K. Liu, C. Shen, B. Wang, S. Fan, J. Zhang, Y. Wu, Z. Wang, and J. Ye. Distribution-aligned sequence distillation for superior long-cot reasoning. *CoRR*, abs/2601.09088, 2026.

[36] L. Yang, Y. Liao, Y. Wang, and Y. Wang. Slora: Balancing plasticity and forgetting in large language models for continual learning. In *Annual Meeting of the Association for Computational Linguistics*, pages 5437–5454, 2026.

[37] S. Yang, K. Ning, Y. Liu, J. Yao, Y. Tian, Y. Song, and L. Yuan. Is parameter collision hindering continual learning in llms? In *International Conference on Computational Linguistics*, pages 4243–4259, 2025.

[38] W. Yu, T. Pang, Q. Liu, C. Du, B. Kang, Y. Huang, M. Lin, and S. Yan. Bag of tricks for training data extraction from language models. In *International Conference on Machine Learning*, pages 40306–40320, 2023.

[39] Y. Zhang, X. Guo, Y. Zhao, X. Sui, B. Zhou, X. Qian, and X. Yuan. Grow-on-demand: Sparse and adaptive expert expansion for continual instruction tuning. In *AAAI Conference on Artificial Intelligence*, pages 28474–28482, 2026.

[40] W. Zhao, S. Wang, Y. Hu, Y. Zhao, B. Qin, X. Zhang, Q. Yang, D. Xu, and W. Che. SAPT: A shared attention framework for parameter-efficient continual learning of large language models. In *Annual Meeting of the Association for Computational Linguistics*, pages 11641–11661, 2024.

A Theoretical Proofs

In this section, we address the following three questions by providing theoretical proofs.

RQ1. *From the data perspective*, can continuous Gumbel-Softmax relaxation be utilized to estimate the pre-training distribution $\mathcal{T}_0$ composed of discrete token sequences?

RQ2. *From the local optimization perspective*, can our proposed orthogonal projection method theoretically resolve the catastrophic forgetting?

RQ3. *From the global generalization perspective*, can maintaining only a soft prompt and its generated surrogate sequences effectively preserve the vast repository of pre-trained knowledge?

A.1 Can Gumbel-Softmax Relaxation Estimate Discrete Pre-training Sequences?

In Sec. 3.2, we utilize a continuous soft prompt $\mathbf{P}_t$ to generate a sequence of soft token embeddings $\mathbf{S}(\mathbf{P}_t)$. However, the authentic pre-training distribution $\mathcal{T}_0$ inherently resides in a discrete text space. We must mathematically guarantee that the stability objective $\mathcal{L}_{\text{pre}}$ evaluated on our differentiable soft sequence effectively bounds the expected loss of a true discrete text sequence, and that this approximation gap strictly vanishes as the Gumbel-Softmax temperature τ approaches zero.

Theorem 1. *As formulated in Sec. 3.2, let $\mathbf{W}_{emb} \in \mathbb{R}^{|\mathcal{V}| \times D}$ be the token embedding matrix. At decoding step m, let $\boldsymbol{\pi}_m(\tau) \in \mathbb{R}^{|\mathcal{V}|}$ be the continuous probability vector generated by the Gumbel-Softmax function at temperature τ. Let $\mathbf{y}_m \in \{0,1\}^{|\mathcal{V}|}$ be the true discrete one-hot vector obtained by an ideal* $\mathbf{argmax}$ *operation on the logits, e.g., greedy sampling. Assume the knowledge distillation loss function $\mathcal{L}_{pre}$ is L-Lipschitz continuous with respect to the input embeddings. The expected gap between the loss evaluated on the soft embedding and the discrete embedding is strictly bounded by $\mathcal{O}(\tau)$, i.e., a linear correlation.*

Proof. We define the continuous soft token embedding as $\mathbf{e}_{\text{soft}} = \boldsymbol{\pi}_m(\tau)\mathbf{W}_{\text{emb}}$ and the corresponding true discrete token embedding as $\mathbf{e}_{\text{discrete}} = \mathbf{y}_m\mathbf{W}_{\text{emb}}$. By the definition of L-Lipschitz continuity [11], the magnitude of the difference in the loss function is bounded by the Euclidean distance between the input vectors multiplied by the Lipschitz constant L, which gives

$$|\mathcal{L}_{\text{pre}}(\mathbf{e}_{\text{soft}}) - \mathcal{L}_{\text{pre}}(\mathbf{e}_{\text{discrete}})| \leq L\|\mathbf{e}_{\text{soft}} - \mathbf{e}_{\text{discrete}}\|_2.$$

We expand this distance metric by substituting the definitions of the respective embeddings:

$$\|\mathbf{e}_{\text{soft}} - \mathbf{e}_{\text{discrete}}\|_2 = \|(\boldsymbol{\pi}_m(\tau) - \mathbf{y}_m)\mathbf{W}_{\text{emb}}\|_2 \leq \|\mathbf{W}_{\text{emb}}\|_{\max} \cdot \|\boldsymbol{\pi}_m(\tau) - \mathbf{y}_m\|_1,$$

where $\|\mathbf{W}_{\text{emb}}\|_{\max}$ denotes the maximum L_2-norm of any single token embedding vector within the vocabulary $\mathcal{V}$. A foundational property of the Gumbel-Softmax relaxation [9] is that the expected L_1 distance between the soft probability distribution $\boldsymbol{\pi}_m(\tau)$ and the hard categorical one-hot vector $\mathbf{y}_m$ scales linearly with the temperature τ as τ becomes small. Consequently, we have $\mathbb{E}[\|\boldsymbol{\pi}_m(\tau) - \mathbf{y}_m\|_1] \leq C \cdot \tau$, where C is a distribution-dependent constant. Taking the expectation on both sides of our Lipschitz inequality, we conclude:

$$\mathbb{E}\big[|\mathcal{L}_{\text{pre}}(\mathbf{e}_{\text{soft}}) - \mathcal{L}_{\text{pre}}(\mathbf{e}_{\text{discrete}})|\big] \leq L \cdot \|\mathbf{W}_{\text{emb}}\|_{\max} \cdot C \cdot \tau = \mathcal{O}(\tau).$$

This mathematically proves that by annealing the temperature $\tau \to 0$ during the generative replay phase, our fully differentiable soft sequence seamlessly and accurately approximates the true discrete text distribution, introducing negligible approximation error while perfectly preserving the gradient flow for optimizing $\mathbf{P}_t$. □

A.2 Can EOUPCT Theoretically Resolve the Catastrophic Forgetting?

We show that our proposed EOUPCT, which builds upon the principles of orthogonal gradient projection, effectively resolves the forgetting as well. In Step 3 of our Pareto orthogonal optimizer in Sec. 3.4, when the historically safe new task gradient $\tilde{\mathbf{g}}_{\text{new}}$ heavily conflicts with the protection gradient $\tilde{\mathbf{g}}_{\text{pre}}$, we project $\tilde{\mathbf{g}}_{\text{new}}$ onto the normal plane of $\tilde{\mathbf{g}}_{\text{pre}}$. We must guarantee that this specific projection is not a heuristic, but the exact geometric optimal solution that minimizes the new task loss while strictly prohibiting any increase in the general knowledge distillation loss.

Theorem 2. *Let $\tilde{\mathbf{g}}_{new}$ and $\tilde{\mathbf{g}}_{pre}$ be the task and protection gradients as Eq. (10), respectively, both already projected into the historical null space via $\mathbf{\Pi}_{new}$. If a gradient conflict is detected, i.e.,*

$\langle \tilde{\mathbf{g}}_{new}, \tilde{\mathbf{g}}_{pre} \rangle < 0$, *the proposed update direction* $\tilde{\mathbf{g}}^*_{new} = \tilde{\mathbf{g}}_{new} - \frac{\langle \tilde{\mathbf{g}}_{new}, \tilde{\mathbf{g}}_{pre} \rangle}{\|\tilde{\mathbf{g}}_{pre}\|^2} \tilde{\mathbf{g}}_{pre}$ *is the exact analytical solution to the constrained optimization problem: maximize the alignment with the new task gradient, subject to absolute zero interference with the estimated pre-training general-purpose knowledge.*

Proof. We formulate the constrained gradient adjustment problem by seeking a modified update direction $\mathbf{d}$ that is geometrically as close as possible to the ideal new task gradient $\tilde{\mathbf{g}}_{\text{new}}$ (minimizing their Euclidean distance), subject to the hard constraint that moving along $\mathbf{d}$ will not increase the distillation loss $\mathcal{L}_{\text{pre}}$. To achieve this, $\mathbf{d}$ must be strictly orthogonal to the protection gradient $\tilde{\mathbf{g}}_{\text{pre}}$. Mathematically, this is expressed as $\mathbf{d}^* = \mathbf{argmin}_{\mathbf{d}}(\|\mathbf{d} - \tilde{\mathbf{g}}_{\text{new}}\|^2/2)$ subject to $\langle \mathbf{d}, \tilde{\mathbf{g}}_{\text{pre}} \rangle = 0$. To solve this constrained objective, we construct the Lagrangian by introducing a Lagrange multiplier $\lambda \in \mathbb{R}$ for the equality constraint:

$$\mathcal{L}(\mathbf{d}, \lambda) = \frac{1}{2}\|\mathbf{d} - \tilde{\mathbf{g}}_{\text{new}}\|^2 + \lambda \langle \mathbf{d}, \tilde{\mathbf{g}}_{\text{pre}} \rangle.$$

To locate the minimum, we compute the derivative of the Lagrangian $\mathcal{L}$ with respect to $\mathbf{d}$ and set it to the zero vector, yielding $\nabla_{\mathbf{d}}\mathcal{L}(\mathbf{d}, \lambda) = (\mathbf{d} - \tilde{\mathbf{g}}_{\text{new}}) + \lambda \tilde{\mathbf{g}}_{\text{pre}} = \mathbf{0}$, which rearranges to $\mathbf{d} = \tilde{\mathbf{g}}_{\text{new}} - \lambda \tilde{\mathbf{g}}_{\text{pre}}$. We then substitute this expression for $\mathbf{d}$ back into the strict hard orthogonality constraint $\langle \mathbf{d}, \tilde{\mathbf{g}}_{\text{pre}} \rangle = 0$:

$$\langle \tilde{\mathbf{g}}_{\text{new}} - \lambda \tilde{\mathbf{g}}_{\text{pre}}, \tilde{\mathbf{g}}_{\text{pre}} \rangle = 0 \implies \langle \tilde{\mathbf{g}}_{\text{new}}, \tilde{\mathbf{g}}_{\text{pre}} \rangle - \lambda \|\tilde{\mathbf{g}}_{\text{pre}}\|^2 = 0.$$

Solving this algebraic equation for the Lagrange multiplier gives $\lambda = \frac{\langle \tilde{\mathbf{g}}_{\text{new}}, \tilde{\mathbf{g}}_{\text{pre}} \rangle}{\|\tilde{\mathbf{g}}_{\text{pre}}\|^2}$. Finally, substituting λ back into the expression for $\mathbf{d}$ yields exactly our proposed projection update rule from Sec. 3.4:

$$\mathbf{d}^* = \tilde{\mathbf{g}}_{\text{new}} - \frac{\langle \tilde{\mathbf{g}}_{\text{new}}, \tilde{\mathbf{g}}_{\text{pre}} \rangle}{\|\tilde{\mathbf{g}}_{\text{pre}}\|^2} \tilde{\mathbf{g}}_{\text{pre}} \equiv \tilde{\mathbf{g}}^*_{\text{new}}.$$

This derivation proves that our projection mechanism is the rigorous geometric Pareto optimum. It maximally preserves the plasticity required for learning $\mathcal{T}_t$ while mathematically enforcing absolute zero catastrophic forgetting of the estimated pre-training distribution. □

A.3 Can Maintaining Only a Soft Prompt Preserve Pre-trained Knowledge?

EOUPCT does not blindly replay the entire infinite pre-training distribution $\mathcal{T}_0$. Instead, we employ adversarial optimization in Eq. (6) to locate a worst-case boundary sequence $\mathbf{S}(\mathbf{P}^*_t)$ and strictly protect it. A critical theoretical question emerges: why does protecting this specific, adversarially selected subset generated by only a soft prompt guarantee the protection of the vast, global pre-training distribution? We mathematically validate this using minimax generalization bounds.

Theorem 3. *Let* $\mathcal{T}_0$ *denote the true global distribution of general-purpose knowledge. Let* $\mathcal{P}$ *define the expressive continuous space of the soft prompt* $\mathbf{P}_t$*. Assuming our adversarial solver successfully identifies the worst-case prompt* $\mathbf{P}^*_t = \mathbf{argmax}_{\mathbf{P}_t \in \mathcal{P}} \mathcal{L}_{pre}(\Delta \mathbf{W}_{virt}; \mathbf{S}(\mathbf{P}_t))$*, and our Pareto-orthogonal update bounds this worst-case loss such that* $\mathcal{L}_{pre}(\Delta \mathbf{W}_{t+1}; \mathbf{S}(\mathbf{P}^*_t)) \leq \epsilon$*, then the expected catastrophic forgetting over the entire true pre-training distribution is strictly bounded from above.*

Proof. We formalize the total catastrophic forgetting across the pre-training LLM's general-purpose knowledge as the expected loss over the true pre-training distribution: $\text{Forgetting}_{\text{global}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{T}_0}[\mathcal{L}_{\text{pre}}(\Delta \mathbf{W}_{t+1}; \mathbf{x})]$. We operate under the standard assumption that the soft prompt space $\mathcal{P}$ is sufficiently expressive such that the generated continuous sequences can smoothly cover the support of the true distribution $\mathcal{T}_0$. Consequently, for any arbitrary actual data sample $\mathbf{x} \in \mathcal{T}_0$, there exists some prompt configuration $\mathbf{P}_{\mathbf{x}} \in \mathcal{P}$ capable of approximating it with an approximation error bounded by a small constant δ, yielding the inequality $\mathcal{L}_{\text{pre}}(\mathbf{x}) \leq \mathcal{L}_{\text{pre}}(\mathbf{S}(\mathbf{P}_{\mathbf{x}})) + \delta$. By the mathematical definition of the maximization operator in our vulnerability objective, the loss computed on any specific generated sequence $\mathbf{S}(\mathbf{P}_{\mathbf{x}})$ must be less than or equal to the loss evaluated on the adversarially optimized, worst-case sequence $\mathbf{S}(\mathbf{P}^*_t)$:

$$\forall \mathbf{x} \in \mathcal{T}_0, \quad \mathcal{L}_{\text{pre}}(\mathbf{S}(\mathbf{P}_{\mathbf{x}})) \leq \max_{\mathbf{P}_t \in \mathcal{P}} \mathcal{L}_{\text{pre}}(\mathbf{S}(\mathbf{P}_t)) = \mathcal{L}_{\text{pre}}(\mathbf{S}(\mathbf{P}^*_t)).$$

By chaining these two inequalities, we elegantly bound the loss of any genuine data point $\mathbf{x}$:

$$\mathcal{L}_{\text{pre}}(\mathbf{x}) \leq \mathcal{L}_{\text{pre}}(\mathbf{S}(\mathbf{P}^*_t)) + \delta.$$

Taking the expectation over $\mathbf{x} \sim \mathcal{T}_0$ strictly preserves this inequality, leading directly to $\mathbb{E}_{\mathbf{x}\sim\mathcal{T}_0}[\mathcal{L}_{\text{pre}}(\mathbf{x})] \leq \mathcal{L}_{\text{pre}}(\mathbf{S}(\mathbf{P}_t^*)) + \delta$. Because our Pareto-orthogonal solver (Sec. 3.4) explicitly enforces that the worst-case loss does not increase (*i.e.,* $\mathcal{L}_{\text{pre}}(\mathbf{S}(\mathbf{P}_t^*)) \leq \epsilon$), the global expected forgetting is naturally bounded by:

$$\text{Forgetting}_{\text{global}} \leq \epsilon + \delta.$$

This mathematically proves that by explicitly hunting for and strictly protecting the vulnerability boundaries (the adversarial "hard negatives"), EOUPCT casts a robust, overarching protective net across the entire global distribution $\mathcal{T}_0$. □

Remark 3.1. A potential critique of Theorem 3 is the assumption that the continuous prompt space $\mathcal{P}$ can densely cover the vast support of the true pre-training distribution $\mathcal{T}_0$. We clarify that our bound does not require global coverage. Instead, we rely on the concept of *local adversarial representativeness*. For a given fine-tuning task at step t, the catastrophic forgetting is primarily driven by the parameter sub-space aligned with the task gradient. Therefore, optimizing $\mathbf{P}_t$ via the virtual step naturally forces the generated sequence $\mathbf{S}(\mathbf{P}_t)$ to act as an adversarial coreset. This coreset only needs to cover the specific local manifold of $\mathcal{T}_0$ that is most vulnerable to the current orthogonal update, bounded by a local approximation error $\delta_{local} \ll \delta_{global}$. As a result, the theoretical upper bound on forgetting remains tight and non-vacuous within the task-specific projection subspace.

B Complexity Analysis of EOUPCT

In this section, we analyze the computational efficiency of the EOUPCT framework. Appendix B.1 mathematically demonstrates why directly solving the multi-objective optimization problem in Eqs. (5), (6), and (7) inherently requires intractable second-order Hessian matrices, and how our efficient Pereto optimizer in Sec. 3.4 reduces this to a first-order optimization; Appendix B.2 provides a step-by-step time complexity comparison between continual fine-tuning baselines and EOUPCT; Appendix B.3 empirically shows the computational budgets of EOUPCT.

B.1 Efficient First-Order Pareto Optimization

The core of our vulnerability objective **[C2]** is to explicitly find a soft prompt $\mathbf{P}_t$ that generates data maximizing the gradient conflict with the new task. Mathematically, this means maximizing the negative inner product between the new task gradient $\mathbf{g}_{\text{new}}$ and the protection gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{pre}}$. We now demonstrate why directly optimizing this requires a Hessian, and how we avoid it.

Second-order optimization bottleneck. Let us formalize the naive objective. To find the worst-case prompt $\mathbf{P}_t$, we want to solve:

$$\max_{\mathbf{P}_t} \mathcal{J}(\mathbf{P}_t) = \big\langle -\mathbf{g}_{\text{new}},\ \nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{pre}}\big(\boldsymbol{\theta};\ \mathbf{S}(\mathbf{P}_t)\big)\big\rangle = -\mathbf{g}_{\text{new}}^{\top}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{pre}}.$$

To optimize this via gradient ascent, we must compute the derivative of $\mathcal{J}(\mathbf{P}_t)$ with respect to the prompt parameters $\mathbf{P}_t$. Since the new task gradient $\mathbf{g}_{\text{new}}$ is computed on $\mathcal{T}_t$ and is strictly independent of $\mathbf{P}_t$, it acts as a constant during this differentiation. Applying the chain rule yields:

$$\nabla_{\mathbf{P}_t}\mathcal{J}(\mathbf{P}_t) = \nabla_{\mathbf{P}_t}\Big[-\mathbf{g}_{\text{new}}^{\top}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{pre}}\Big] = -\Big(\nabla^2_{\mathbf{P}_t,\boldsymbol{\theta}}\mathcal{L}_{\text{pre}}\Big)\mathbf{g}_{\text{new}}.$$

The term $\nabla^2_{\mathbf{P}_t,\boldsymbol{\theta}}\mathcal{L}_{\text{pre}}$ is a *mixed second-order derivative matrix*, *i.e.,* a Hessian cross-derivative. Computing a Hessian-vector product through the massive layers of an LLM requires unrolling the entire computational graph of the first gradient backward. This triggers an $\mathcal{O}(N^2)$ memory explosion and massive computational latency, making it absolutely intractable for LLMs.

First-order resolution of EOUPCT. Instead of differentiating the inner product, EOUPCT employs a *virtual step approximation*. According to the first-order Taylor expansion of $\mathcal{L}_{\text{pre}}$ around the current weights $\boldsymbol{\theta}_t$ with a step size α:

$$\mathcal{L}_{\text{pre}}(\boldsymbol{\theta}_t - \alpha\mathbf{g}_{\text{new}}) \approx \mathcal{L}_{\text{pre}}(\boldsymbol{\theta}_t) - \alpha\mathbf{g}_{\text{new}}^{\top}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{pre}}(\boldsymbol{\theta}_t).$$

Notice that the term $-\mathbf{g}_{\text{new}}^{\top}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{pre}}$ is exactly our desired gradient conflict objective $\mathcal{J}(\mathbf{P}_t)$. By defining the virtual weights as $\boldsymbol{\theta}_{\text{virt}} = \boldsymbol{\theta}_t - \alpha\mathbf{g}_{\text{new}}$, the maximization problem becomes:

$$\max_{\mathbf{P}_t} \mathcal{L}_{\text{pre}}\big(\boldsymbol{\theta}_{\text{virt}};\ \mathbf{S}(\mathbf{P}_t)\big).$$

To optimize this new objective, we simply take the derivative w.r.t. $\mathbf{P}_t$:

$$\nabla_{\mathbf{P}_t} \mathcal{L}_{\text{pre}}\big(\boldsymbol{\theta}_{\text{virt}};\, \mathbf{S}(\mathbf{P}_t)\big).$$

Because $\boldsymbol{\theta}_{\text{virt}}$ is treated as a fixed parameter matrix during this step, this operation is a *standard, first-order backpropagation*. We completely bypass the Hessian matrix, reducing the spatial and temporal complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, rendering the adversarial training highly scalable.

B.2 Time Complexity Compared with Standard LoRA Baselines

We now quantify the exact computational overhead introduced by EOUPCT compared to standard LoRA fine-tuning. We define the time complexity of a standard LLM forward pass as $\mathcal{O}(\text{FP}_{\text{LLM}})$, a backward pass to update LoRA weights as $\mathcal{O}(\text{BP}_{\text{LoRA}})$, and a backward pass to update the soft prompt as $\mathcal{O}(\text{BP}_{\text{Prompt}})$. Note that $\text{BP}_{\text{LoRA}} \approx \text{BP}_{\text{Prompt}} \ll \text{BP}_{\text{Full_Model}}$.

In a standard LoRA fine-tuning iteration, the model processes a batch of new task data, computes the loss, and backpropagates to update the LoRA parameters, *i.e.,*$\mathcal{O}(\text{FP}_{\text{LLM}}) + \mathcal{O}(\text{BP}_{\text{LoRA}})$.

Turn to analyze EOUPCT, a single iteration of EOUPCT performs the following sequential steps:

1. **New task gradient (*Step 1*):** Compute $\mathbf{g}_{\text{new}}$ on $\mathcal{T}_t$, *i.e.,* $\mathcal{O}(\text{FP}_{\text{LLM}}) + \mathcal{O}(\text{BP}_{\text{LoRA}})$;
2. **Pre-training gradient estimation (*Sec. 3.2*):** Auto-regressively generating a soft sequence of length M using Gumbel-Softmax requires generating M tokens iteratively, so the cost is $M \times \mathcal{O}(\text{FP}_{\text{LLM}})$;
3. **Adversarial prompt update (*Step 1*):** Evaluate $\mathcal{L}_{\text{pre}}$ using the generated sequence on $\boldsymbol{\theta}_{\text{virt}}$, and backpropagate to update $\mathbf{P}_t$, *i.e.,* $\mathcal{O}(\text{FP}_{\text{LLM}}) + \mathcal{O}(\text{BP}_{\text{Prompt}})$;
4. **Protection gradient calculation (*Step 1*):** Evaluate $\mathcal{L}_{\text{pre}}$ again on the true weights $\boldsymbol{\theta}_t$ to obtain $\mathbf{g}^*_{\text{pre}}$, *i.e.,* $\mathcal{O}(\text{FP}_{\text{LLM}}) + \mathcal{O}(\text{BP}_{\text{LoRA}})$;
5. **Pareto orthogonal projection (*Steps 2 & 3*):** Project the gradients using the historical memory matrix $\mathbf{M}_{\text{new}}$, which contains $t-1$ vectors of dimension D_{LoRA}. This involves basic matrix-vector multiplications on the parameter space, which is negligible compared to passing tensors through the LLM layers. Therefore, its cost is $\mathcal{O}(D_{\text{LoRA}} \cdot t) \approx \mathcal{O}(1)$ relative to LLM passes.

Summing the costs, a single step of EOUPCT requires $\mathcal{O}\big((M+3)\text{FP}_{\text{LLM}} + 2\text{BP}_{\text{LoRA}} + \text{BP}_{\text{Prompt}}\big)$. Because the length of the surrogate sequence M is typically a small constant, the overall complexity of EOUPCT scales *linearly* with the LLM size, identically to standard LoRA. We successfully mitigate the catastrophic forgetting by introducing only a *small constant-factor computational overhead*, completely avoiding the exponential $\mathcal{O}(N^2)$ scaling that traditionally plagues orthogonal gradient alignment and adversarial generation in LLMs.

B.3 Empirical Computational Budget

In Table 3, we present an empirical comparison of the computational time required by our proposed method, EOUPCT, against standard LoRA and other advanced continual LLM fine-tuning baselines, including LoRAMoE [3] and CLoRA [17]. All experiments are measured in running minutes using an 8×GPUs distributed training setup across various representative LLM architectures (Qwen3, Llama3, and Gemma2 families). As expected, standard LoRA serves as the lower bound for training time (averaging 60 minutes, denoted as $1\times$) due to its straightforward low-rank weight injection. However, methods designed to enhance model capacity or mitigate catastrophic forgetting inherently introduce substantial computational overhead. For instance, CLoRA aims to reduce output variations and preserve previous knowledge by applying subspace regularization, which imposes strict mathematical constraints on the direction of the updating matrix's null space. While effective for continual learning, this regularization mechanism significantly burdens the training process, resulting in an average computational cost of $2.6\times$ compared to standard LoRA. Similarly, LoRAMoE incurs the highest computational budget (averaging $3.2\times$) due to the complex routing mechanisms and multiple expert networks involved during the forward and backward passes.

In contrast, our proposed EOUPCT demonstrates superior computational efficiency among the advanced fine-tuning methods. Across all evaluated model scales, EOUPCT consistently requires considerably less training time than both CLoRA and LoRAMoE. On average, EOUPCT only takes

Table 3: Empirical computational time (minutes) on 8×GPUs.

LLM	LoRA	LoRAMoE *(#expert=4)*	LoRAMoE *(#expert=8)*	OLoRA	CLoRA	EOUPCT
Qwen3-8B	88 (1×)	223 (2.5×)	225 (2.5×)	237 (2.7×)	191 (2.2×)	162 (1.8×)
Qwen3-4B	62 (1×)	194 (3.1×)	195 (3.1×)	204 (3.3×)	155 (2.5×)	129 (2.1×)
Llama3-8B	60 (1×)	197 (3.3×)	199 (3.3×)	211 (3.5×)	156 (2.6×)	130 (2.1×)
Llama3-3B	35 (1×)	144 (4.1×)	144 (4.1×)	151 (4.3×)	114 (3.2×)	65 (1.8×)
Gemma2-9B	76 (1×)	260 (3.4×)	262 (3.4×)	271 (3.5×)	210 (2.7×)	178 (2.3×)
Gemma2-2B	44 (1×)	140 (3.2×)	141 (3.2×)	149 (3.4×)	116 (2.6×)	69 (1.5×)
Average	60 (1×)	193 (3.2×)	194 (3.2×)	203 (3.4×)	157 (2.6×)	122 (2.0×)

Table 4: Summary of primary notations used in EOUPCT.

Notation	**Description**
$\mathcal{F}_{\boldsymbol{\theta}}, \boldsymbol{\theta}$	Frozen pre-training LLM and trainable LLM weights
$\mathcal{T}_t, \mathcal{T}_0$	t-th new downstream task and original pre-training distribution
$\mathbf{P}_t, \mathbf{S}(\mathbf{P}_t)$	Continuous soft prompt and generated surrogate sequence
$\mathcal{L}_{\text{new}}, \mathcal{L}_{\text{pre}}$	Plasticity and stability objective functions
$\mathbf{g}_{\text{new}}, \mathbf{g}_{\text{pre}}$	Task-specific and protection gradients w.r.t. $\boldsymbol{\theta}$
$\boldsymbol{\theta}_{\text{virt}}$	Virtually updated weights simulating new task impact
$\mathcal{M}_{\text{new}}, \mathbf{\Pi}_{\text{new}}$	Historical gradient memory bank and its null-space projection matrix
$\tilde{\mathbf{g}}_{\text{new}}, \tilde{\mathbf{g}}_{\text{pre}}$	Gradients projected into historically safe null space
$\tilde{\mathbf{g}}^*_{\text{new}}$	Pareto-optimal modified new task gradient
η, α, λ	learning rate, virtual step size, and orthogonal penalty coefficient

$2.0\times$ the time of standard LoRA, saving roughly 23% and 37% of the training time compared to CLoRA and LoRAMoE, respectively. Notably, the efficiency of EOUPCT becomes even more pronounced on relatively compact models; for example, on Llama3-3B and Gemma2-2B, the relative computational cost drops to just $1.8\times$ and $1.5\times$. These results indicate that EOUPCT successfully bypasses the severe computational bottlenecks typically associated with strict orthogonal gradient projection (as seen in CLoRA) or dense mixture-of-experts routing (as in LoRAMoE). Consequently, EOUPCT offers a highly practical and scalable solution, striking an optimal balance between advanced model adaptability and empirical training efficiency.

C Algorithm and Notations

In this section, we summarize the notations used in our proposed method and provide the complete algorithmic workflow of EOUPCT. Table 4 summarizes the primary mathematical notations and their corresponding descriptions used throughout the formulation of our method. As analyzed in Sec. 3.4, directly solving the multi-objective optimization problem for continual fine-tuning is computationally intractable due to the natural adversariality and the requirement of second-order Hessian matrices. To address this, EOUPCT elegantly decouples the complex optimization into an efficient first-order bilevel training paradigm. The overall training procedure is outlined in Alg. 1. For each new task, we first utilize a virtual step approximation to adversarially optimize the soft prompt, extracting the most vulnerable pre-training data distribution. Subsequently, we leverage null-space projection and Pareto-optimal orthogonal projection to eliminate gradient conflicts. This ensures that the newly updated large model efficiently acquires new task plasticity while strictly maintaining general-purpose knowledge stability and retaining previous downstream task abilities.

D Experimental Setups

To rigorously evaluate EOUPCT under a continual fine-tuning setting, we conduct experiments on a broad suite of multi-task benchmarks and multiple open-source pre-training LLMs. In this section, we describe the pre-training LLMs, datasets, baseline methods, evaluation metrics, and implementation details used throughout our experiments.

Algorithm 1 Training Process of EOUPCT

Require: Frozen pre-training LLM $\mathcal{F}_{\boldsymbol{\theta}}$, task sequence $\{\mathcal{T}_1, \ldots, \mathcal{T}_T\}$, learning rates η, α, penalty coefficient λ.
Ensure: Optimized LLM parameters $\boldsymbol{\theta}$.

Initialize trainable LLM parameters $\boldsymbol{\theta}$ with $\boldsymbol{\theta}^0$.
Initialize historical gradient memory sets $\mathcal{M}_{\text{new}} \leftarrow \varnothing$, $\mathcal{M}_{\text{pre}} \leftarrow \varnothing$.
for each new task $t = 1, 2, \ldots, T$ **do**
  Initialize the learnable soft prompt $\mathbf{P}_t$.
  for each training iteration **do**
    Sample a mini-batch of data from $\mathcal{T}_t$.
    Compute new task gradient $\mathbf{g}_{\text{new}} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{new}}$.
    ▷ *Step 1: Vulnerability & pre-training gradient estimation*
    Perform virtual update $\boldsymbol{\theta}_{\text{virt}}$ via Eq. (8).
    Optimize $\mathbf{P}_t$ by maximizing Eq. (6) to generate $\mathbf{S}(\mathbf{P}_t^*)$.
    Compute protection gradient $\mathbf{g}_{\text{pre}}^* = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{pre}}(\boldsymbol{\theta}; \mathbf{S}(\mathbf{P}_t^*))$.
    ▷ *Step 2: Null-space projection*
    Construct projection matrix $\mathbf{\Pi}_{\text{new}}$ via Eq. (9) using $\mathcal{M}_{\text{new}}$.
    Obtain safe gradients $\tilde{\mathbf{g}}_{\text{new}} = \mathbf{\Pi}_{\text{new}} \mathbf{g}_{\text{new}}$ and $\tilde{\mathbf{g}}_{\text{pre}} = \mathbf{\Pi}_{\text{new}} \mathbf{g}_{\text{pre}}^*$.
    ▷ *Step 3: Pareto orthogonal optimization*
    if $\langle \tilde{\mathbf{g}}_{\text{new}}, \tilde{\mathbf{g}}_{\text{pre}} \rangle < 0$ **then**
      Compute interference-free gradient $\tilde{\mathbf{g}}_{\text{new}}^*$ via Pareto projection in Eq. (11).
    else
      Set $\tilde{\mathbf{g}}_{\text{new}}^* = \tilde{\mathbf{g}}_{\text{new}}$.
    end if
    ▷ *Step 4: Parameter update*
    Update LLM parameters $\boldsymbol{\theta}$ following the direction in Eq. (12).
  end for
  Update memory: $\mathcal{M}_{\text{new}} \leftarrow \mathcal{M}_{\text{new}} \cup \{\mathbf{g}_{\text{new}}\}$, $\mathcal{M}_{\text{pre}} \leftarrow \mathcal{M}_{\text{pre}} \cup \{\mathbf{g}_{\text{pre}}^*\}$.
end for
return $\boldsymbol{\theta}$

D.1 Pre-training LLMs

To comprehensively evaluate the effectiveness of our method, we conduct experiments on three prominent families of open-weight decoder-only LLMs, selecting two parameter scales for each family: **Qwen3** (**4B** and **8B**) [26], **Llama3** (**3B** and **8B**) [1], and **Gemma2** (**2B** and **9B**) [25]. These LLM families provide a diverse and representative testbed spanning different pre-training corpora, architectural variations, and capacity regimes:

- *The Qwen3 family* [26] is renowned for its exceptional multilingual capabilities and robust alignment with human instructions. We include the 4B and 8B variants to investigate continual learning dynamics, which distill from their larger-scale variant, *e.g.,* Qwen3-32B, by the off-policy and on-policy distillation.
- *The Llama3 family* [1] is pre-trained on a massive and highly curated corpus of over 9 trillion tokens. It features an optimized Transformer architecture with grouped-query attention and an expanded vocabulary size. We utilize the 3B and 8B variants to represent highly capable models at accessible scales.
- *The Gemma2 family* [25] introduces architectural innovations such as interleaved local and global attention mechanisms and logit soft-capping. The 2B and 9B models are selected to evaluate our framework's stability on highly optimized, dense architectures.

By evaluating across these distinct families and scales, we aim to demonstrate that our continual fine-tuning framework is robust and generally applicable regardless of the underlying LLM architecture.

D.2 Benchmark Datasets

We evaluate our continual LLM fine-tuning framework on two complementary benchmarks, *SuperNI* and *MMLU*, aiming to jointly assess task-specific adaptation and general-purpose knowledge

Table 5: Overview of selected multi-task datasets from *SuperNI* [32].

ID	Dataset	Task Type	Train	Eval
t1290	xsum_summarization	text summarization	1,000	100
t511	reddit_tifu_long_text_summarization	text summarization	1,000	100
t1572	samsum_summary	text summarization	160	20
t1729	personachat_generate_next	dialogue generation	1,000	100
t639	multi_woz_user_utterance_generation	dialogue generation	142	18
t1590	diplomacy_text_generation	dialogue generation	126	16
t1687	sentiment140_classification	sentiment analysis	1,000	100
t363	sst2_polarity_classification	sentiment analysis	1,000	100
t875	emotion_classification	sentiment analysis	1,000	100
t748	glucose_reverse_cause_event_detection	information extraction	1,000	100
t1510	evalution_relation_extraction	information extraction	1,000	100
t181	outcome_extraction	information extraction	338	43
t591	sciq_answer_generation	question answering	1,000	100
t002	quoref_answer_generation	question answering	1,000	100
t073	commonsenseqa_answer_generation	question answering	975	100

Table 6: Three distinct task orderings under the continual LLM fine-tuning.

Seed	Task Ordering
1	t591 → t073 → t1687 → t875 → t1572 → t639 → t1510 → t1590 → t511 → t181 → t363 → t1729 → t002 → t1290 → t748
2	t1290 → t1510 → t1572 → t002 → t073 → t511 → t1590 → t363 → t748 → t639 → t181 → t1729 → t1687 → t591 → t875
3	t002 → t073 → t1572 → t181 → t591 → t1687 → t363 → t1729 → t511 → t875 → t639 → t748 → t1590 → t1290 → t1510

preservation under a continual training stream. For continual task learning, we adopt a subset of the *SuperNI* benchmark [32], consisting of 15 diverse tasks spanning five categories, including question answering, information extraction, sentiment analysis, text summarization, and dialogue generation, as summarized in Table 5. Each task is associated with its own training and evaluation splits, and the LLM is trained sequentially over the task stream, where each task is observed only once. After completing the training on each task, we evaluate the LLM on the evaluation sets of all tasks, enabling a unified assessment of their accuracy and forgetting rate on previously learned tasks. To mitigate the bias induced by a specific task sequence, we further consider three distinct task orderings generated from different random seeds, as shown in Table 6, and conduct all experiments independently on each ordering to ensure robustness against ordering effects.

In addition to task-specific performance, we evaluate the LLM's general-purpose knowledge using a subset of the *MMLU* benchmark [6], which covers a broad range of domains including STEM, humanities, and other disciplines, as listed in Table 7. Following the standard *MMLU* protocol, the development split is used to construct 5-shot contexts, while performance is measured on the test split. Importantly, consistent with our continual learning setup, we interleave evaluation with training by testing the LLM on the full *MMLU* benchmark after each task update. This design allows us to track the evolution of general-purpose knowledge throughout the continual fine-tuning process, thereby providing a comprehensive view of the trade-off between task specialization and generalization in LLMs.

Table 7: Statistic of *MMLU* subjects.

Subject	Category	#Eval
anatomy	other	135
global_facts	other	100
marketing	other	234
philosophy	humanities	311
formal_logic	humanities	283
prehistory	humanities	324
high_school_math	STEM	270
high_school_physics	STEM	154
high_school_CS	STEM	287

D.3 Baselines Methods

We compare our method with several representative continual LLM fine-tuning approaches as follows:

- **LoRA** [7] is a widely adopted method that isolates and reparameterizes the weight updates of a pre-trained LLM into low-rank matrices. By freezing the backbone parameters and only optimizing the low-rank adapters, LoRA significantly reduces the number of trainable parameters.
- **LoRAMoE** [3] extends LoRA by incorporating a mixture-of-experts mechanism into the low-rank adaptation process. Specifically, multiple LoRA experts are maintained, and a routing strategy dynamically selects or combines these experts for each input, thereby improving model capacity and adaptability across tasks.
- **O-LoRA** [31] introduces orthogonality constraints on the LoRA update matrices to mitigate interference between tasks. By encouraging the updates of different tasks to lie in orthogonal subspaces, O-LoRA reduces catastrophic forgetting while preserving the efficiency of low-rank adaptation.
- **GainLoRA** [15] further enhances the LoRA-based continual learning paradigm by introducing a gated integration mechanism over multiple LoRA branches. For each new task, a new LoRA branch is added, and learnable gating modules are used to control the contribution of each branch. This design effectively suppresses the influence of newly learned parameters on previous tasks, thereby alleviating forgetting without requiring task identity during inference.
- **CLoRA** [17] is a learning-based method that imposes subspace regularization on LoRA updates. Instead of only constraining the rank of the update, CLoRA further restricts the direction of the update matrix by enforcing orthogonality with respect to a predefined subspace. This strategy reduces the magnitude of output perturbations while maintaining sufficient model capacity, achieving a better trade-off between adaptation and forgetting.

These baselines collectively cover standard LoRA fine-tuning variants and state-of-the-art continual fine-tuning strategies, providing a comprehensive comparison for evaluating our method.

D.4 Evaluation Metrics

To rigorously assess both task-specific adaptation and the preservation of previously acquired general-purpose knowledge during continual fine-tuning, we define specific performance and forgetting metrics for the *SuperNI* and *MMLU* benchmarks.

***SuperNI* metrics.** Given that *SuperNI* comprises diverse generation tasks, we utilize the standard ROUGE score (specifically ROUGE-L) to measure the generation quality by computing the longest common subsequence overlap between the LLM's prediction and the ground-truth reference. To evaluate the stability of the LLM against catastrophic forgetting on sequential tasks, we introduce the *forgetting rate* (Fgt.). For a continually fine-tuned LLM, let $R_{t,t}$ denote the ROUGE score evaluated on task t's test set immediately after the LLM finishes training on task t. Let $R_{T,t}$ denote the ROUGE score on task t after the LLM has sequentially finished training on all T tasks. The forgetting rate for task t is defined as the performance drop between these two stages:

$$\mathrm{FR}_t = R_{t,t} - R_{T,t}.$$

The overall forgetting rate on *SuperNI* is calculated as the average of FR_t across all T learned tasks: $\mathrm{FR}_{\textit{SuperNI}} = \frac{1}{T}\sum_{t=1}^{T}\mathrm{FR}_t$. A lower forgetting rate indicates that the model better retains the knowledge of previously learned tasks.

***MMLU* metrics.** Since *MMLU* is a multi-choice selection dataset designed to test general-purpose knowledge, we use *accuracy* as the primary metric. The overall accuracy is computed by pooling all evaluated questions across the selected subjects. To measure the degradation of the LLM's pre-trained knowledge after adapting to the sequence of downstream tasks, we define the *MMLU* forgetting rate. Let A_0 denote the *MMLU* accuracy of the original pre-training LLM before any continual fine-tuning. Let A_T denote the MMLU accuracy of the LLM after it has been trained on all T tasks in the continual learning stream. The general-purpose knowledge forgetting rate is defined as:

$$\mathrm{FR}_{\textit{MMLU}} = A_0 - A_T.$$

This metric explicitly quantifies the trade-off between acquiring new task-specific skills and preserving the pre-training model's broad, general-purpose knowledge.

Table 8: Performance dynamics during continual fine-tuning, using the **Qwen3-4B** model with the **EOUPCT** method (ROUGE = 50.7, Forget Rate = 6.0).

		Evaluation Task ID														
		591	073	1687	875	1572	639	1510	1590	511	181	363	1729	002	1290	748
Training Task ID	591	67.2	80.1	42.5	41.0	32.6	8.2	24.2	11.2	14.8	13.9	60.6	10.6	77.1	18.8	42.9
	073	68.9	81.0	61.4	45.0	32.0	7.2	15.3	13.7	14.3	13.8	92.0	10.5	78.2	19.1	42.8
	1687	66.8	77.0	86.0	42.0	32.4	7.3	54.4	13.6	14.4	11.6	91.0	10.9	77.6	18.7	43.9
	875	62.9	77.0	85.0	81.0	35.2	6.6	74.8	12.0	13.4	23.8	84.0	10.2	73.8	20.7	43.3
	1572	61.0	67.0	86.0	74.0	40.7	7.0	85.9	10.5	14.8	29.7	87.0	14.1	73.5	18.2	44.5
	639	52.4	59.0	82.0	78.0	35.1	11.6	73.5	12.3	14.1	29.3	84.0	12.8	69.4	18.9	41.1
	1510	60.7	65.0	82.0	78.0	38.1	8.5	100.0	14.0	14.5	36.3	86.0	13.7	73.1	19.6	43.0
	1590	54.1	70.0	82.0	74.0	28.8	4.1	99.7	10.8	13.7	29.3	80.0	10.0	69.6	18.3	41.8
	511	52.2	75.0	76.0	73.0	24.5	4.5	99.3	6.9	14.9	29.9	87.0	6.9	70.0	20.5	40.2
	181	60.0	70.0	83.0	70.0	18.9	1.6	99.7	7.8	15.2	71.3	84.0	6.7	72.6	18.9	41.4
	363	58.8	73.0	82.0	72.0	23.3	6.2	98.2	7.7	15.3	72.1	87.0	8.6	74.5	19.4	41.8
	1729	57.8	74.0	82.0	70.0	24.1	8.9	97.8	11.1	13.6	68.9	89.0	14.6	75.4	21.4	41.2
	002	59.8	71.0	80.0	68.0	29.7	6.2	98.8	9.3	15.2	66.1	90.0	15.3	78.6	19.9	44.3
	1290	59.1	66.0	78.0	70.0	18.6	8.4	94.2	7.5	10.3	69.0	87.0	14.4	77.1	23.5	35.4
	748	58.8	70.0	74.0	68.0	30.5	7.8	92.8	11.9	15.6	69.2	88.0	14.1	71.9	23.3	64.8

D.5 Implementation Details

Our experiments are implemented in PyTorch based on Hugging Face `Transformers` and `PEFT`. We fine-tune the backbone LLMs with LoRA while keeping the original pre-training LLM parameters frozen. During fine-tuning, gradient checkpointing is enabled to lower activation memory consumption, and the input sequence length is truncated to 1024 tokens with the maximum generation length set to 50 tokens. We optimize the trainable parameters with AdamW using an initial learning rate of 2×10^{-4}. To ensure reproducibility, the task order is shuffled once per run under a fixed random seed, and evaluation is performed after each task to track both task acquisition and forgetting over the course of continual learning. For *MMLU* evaluation, we follow the standard 5-shot protocol and set the maximum context length to 2048 tokens. If the composed prompt exceeds the context budget, the number of demonstrations is reduced until the input fits within the limit. During inference, the LLM predicts among the four candidate answers by computing the logits at the final token position and selecting the option with the highest probability. We use batch size 1 for MMLU to preserve the integrity of the few-shot prompt, and this evaluation is run in the same continual-learning pipeline as *SuperNI* to obtain a consistent estimate of both in-domain adaptation and out-of-domain retention.

Stable optimization of pre-training gradient estimation. As introduced in Sec. 3.2, EOUPCT employs the Gumbel-Softmax relaxation to maintain a differentiable computational graph across auto-regressive decoding steps. However, applying Gumbel-Softmax over the entire vocabulary $\mathcal{V}$ (typically over 30k tokens for modern LLMs) across a long sequence can lead to severe memory consumption and gradient instability (*e.g.,* vanishing or exploding gradients). To address this, we introduce two engineering adaptations. First, we restrict the surrogate sequence length to a moderate size (*e.g.,* $M = 16$ or 32). Empirical observations suggest that the model does not need to generate excessively long context to expose general-purpose knowledge vulnerabilities; a short, highly-targeted surrogate sequence $\mathbf{S}(\mathbf{P}_t)$ optimized via the virtual step is entirely sufficient to anchor the general-purpose knowledge boundaries. Second, to alleviate the memory overhead during the softmax expectation, we apply a dynamic vocabulary truncation strategy: at each generation step m, we only compute the Gumbel-Softmax over the top-50 highest probability tokens in $\mathbf{z}_m$, effectively zeroing out the long-tail noise and drastically reducing the size of the differentiable computational graph.

E Performance Dynamics on *SuperNI*

In Tables 8 and 9, we compare the performance dynamics of our approach against the baseline methods across each task in *SuperNI*, recorded sequentially after the training of every task. Our results demonstrate that our method outperforms the baselines in the vast majority of scenarios.

Table 9: Performance dynamics during continual fine-tuning, using the **Qwen3-4B** model with the **GainLoRA** method (ROUGE = 48.4, Forget Rate = 9.0).

		Evaluation Task ID														
		591	073	1687	875	1572	639	1510	1590	511	181	363	1729	002	1290	748
Training Task ID	591	67.8	79.0	78.0	43.0	34.3	4.0	95.7	12.8	14.6	12.3	88.0	11.9	77.1	19.5	43.6
	073	66.3	78.0	75.0	42.0	35.2	9.4	20.2	10.3	14.9	14.2	91.0	10.4	76.9	19.2	43.3
	1687	66.7	80.0	86.0	38.0	34.7	6.3	89.9	11.4	14.9	16.1	89.0	11.9	76.8	18.9	43.5
	875	61.1	66.0	82.0	83.0	37.2	5.3	92.7	14.0	13.9	6.2	79.0	11.4	75.2	18.3	42.6
	1572	55.2	69.0	80.0	77.0	44.8	4.3	93.3	9.1	14.5	28.2	82.0	15.9	76.5	18.7	43.6
	639	52.6	63.0	77.0	74.0	36.0	10.5	84.1	14.2	13.4	36.1	80.0	13.6	77.3	17.0	43.2
	1510	54.9	65.0	75.0	76.0	39.7	12.3	98.2	13.4	14.2	26.6	83.0	13.9	73.1	17.5	43.9
	1590	51.6	66.0	76.0	77.0	31.5	9.6	99.0	12.0	12.6	27.8	85.0	10.9	72.6	16.9	43.0
	511	50.9	66.0	81.0	72.0	28.3	4.8	98.3	7.0	14.9	29.7	84.0	8.7	73.5	20.2	37.5
	181	52.8	64.0	77.0	73.0	30.9	5.5	99.0	5.6	13.9	75.3	80.0	9.7	76.2	19.4	35.9
	363	50.8	65.0	74.0	70.0	21.0	4.3	99.7	5.5	13.8	64.1	91.0	11.1	76.0	18.9	36.7
	1729	51.1	66.0	79.0	64.0	27.8	6.5	99.0	8.0	12.4	67.3	89.0	13.7	77.9	18.7	25.2
	002	43.4	71.0	78.0	62.0	29.4	7.9	99.0	8.9	13.6	61.4	90.0	14.7	78.8	18.4	28.0
	1290	50.3	69.0	75.0	60.0	15.0	6.7	92.2	9.5	8.3	67.9	90.0	13.2	76.1	24.6	22.3
	748	48.1	67.0	76.0	54.0	25.9	7.5	99.0	11.9	10.8	62.4	89.0	13.9	71.4	24.3	65.3

F Limitations

While EOUPCT effectively addresses the catastrophic forgetting in continual LLM fine-tuning without requiring access to original pre-training data, it possesses a few limitations that warrant future exploration. First, the differentiable generation mechanism naturally introduces additional computational overhead during the training phase. Although EOUPCT completely circumvents the massive storage burden of maintaining historical pre-training datasets, auto-regressively generating pseudo data via Gumbel-Softmax increases the latency of the forward and backward passes compared to standard LoRA methods. We provide a comprehensive computational complexity analysis in Sec. B and introduce several implementation refinements to minimize the overhead in Sec. D.5.

Second, the representational capacity of the generated pseudo data sequence is inherently bounded by the predefined length of the soft prompts L and generation steps M, and we provide a sensitivity experiment in Sec. 4.2. While our adversarial virtual step approximation guarantees that these sequences precisely target the most vulnerable boundaries of general-purpose knowledge, dealing with extremely long and highly heterogeneous task sequences might eventually saturate this representational capacity. Future work could explore dynamically expanding prompt lengths or incorporating hierarchical prompt routing to further scale the generation mechanism.

G LLM Usage

LLMs were utilized solely for grammatical refinement and linguistic polishing of the manuscript; the core ideas, methodology, and results were developed independently by the authors.

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: *Our primary motivation is the observation that orthogonal gradient projection methods fail to preserve general-purpose knowledge due to unknown pre-training gradients. Therefore, we introduce* EOUPCT *to estimate and orthogonalize these unknown pre-training gradients. These contributions are detailed extensively in our Abstract and Introduction.*

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: *In Appendix F, we provide a thorough discussion regarding the potential limitations of this work, focusing primarily on computational efficiency, which remains a critical area for prospective investigation.*

   Guidelines:

   - The answer [N/A] 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: *In Appendix A, we provide rigorous theoretical proofs for the claims and underlying assumptions, specifically addressing the three primary theoretical questions, presented throughout this work.*

Guidelines:

- The answer [N/A] 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: *In Sec. 4, we present comprehensive experimental results and analyses. Furthermore, Appendix D is dedicated to detailing the essential information of our experiments, including LLM architectures, baseline methods, datasets, and implementation details.*

Guidelines:

- The answer [N/A] 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 have provided the source code for our implementation, accessible via the link included in Sec. 1.*

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: *Comprehensive details regarding the experimental setup are provided in Appendix D. Furthermore, we perform a sensitivity analysis in Sec. 4.2 to delineate the selection process for key hyper-parameters,* e.g., L *and* M.

   Guidelines:

   - The answer [N/A] 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: *The experiments presented in Table 1 were conducted across three independent trials using distinct random seeds, each corresponding to a different task order. We report the aggregated results from these trials, which consistently demonstrate statistical significance.*

Guidelines:

- The answer [N/A] 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: *Appendix B.3 provides a comprehensive breakdown of the computational runtime measured across our 8-GPU cluster.*

   Guidelines:

   - The answer [N/A] 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 thoroughly reviewed the Code of Ethics and certify that our research adheres to all prescribed ethical standards and institutional guidelines.*

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: *This research focuses on the continual LLM fine-tuning, utilizing exclusively open-source instruction-tuning datasets, such as those for sentiment classification and text summarization. Consequently, this work does not present direct negative societal impacts regarding data privacy or sensitive information.*

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: *Our experiments are conducted using exclusively publicly available datasets and open-source LLMs. Consequently, this work presents no foreseeable risk regarding the misuse of models or datasets, as all resources are utilized in accordance with their respective open-source licenses and intended research purposes.*

Guidelines:

- The answer [N/A] 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 datasets and LLMs employed in this study are open-source, and their respective origins have been duly cited. Consequently, there are no unresolved licensing issues, as our use of these resources strictly adheres to their original distribution terms.*

Guidelines:

- The answer [N/A] 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 have open-sourced the implementation code for our experiments and provided comprehensive documentation regarding its usage.*

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: *Our experiments do not require any additional crowdsourced data collection; all analyses are conducted using existing, publicly available benchmarks.*

Guidelines:

- The answer [N/A] 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: [N/A]

    Justification: *Our experiments do not require any additional crowdsourced data collection; all analyses are conducted using existing, publicly available benchmarks.*

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: *In Appendix G, we disclose that LLMs were employed strictly for grammatical proofreading and linguistic refinement to enhance the readability of the manuscript.*

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.