Unveiling and Addressing Pseudo Forgetting in Large Language Models

Anonymous ACL submission

Abstract

Although substantial efforts have been made to mitigate catastrophic forgetting in continual learning, the intrinsic mechanisms are not well understood. In this work, we demonstrate the existence of "pseudo forgetting": the performance degradation on previous tasks is not attributed to a loss of capabilities, but rather to the failure of the instructions to activate the appropriate model capabilities. We show that the model's performance on previous tasks can be restored through two simple interventions: (1) providing partial external correct rationale, and (2) appending semantically meaningless suffixes to the original instructions, to guide the generation of correct rationales. Through empirical analysis of the internal mechanisms governing rationale generation, we reveal that models exhibiting pseudo forgetting show reduced instruction dependence during rationale generation, leading to suboptimal activation of their inherent capabilities. Based on this insight, we propose Rationale-Guidance Difficulty based Replay (RGD-R) framework that dynamically allocates replay data based on the model's ability to correctly leverage the intrinsic capabilities. Experimental results demonstrate that RGD-R effectively mitigates pseudo forgetting while maintaining model plasticity.

1 Introduction

Continual learning enables Large Language Models (LLMs) (Brown et al., 2020; Yang et al., 2023) to incrementally learn from a sequence of tasks, helping LLMs adapt to the dynamic nature of real-world data and improve their capabilities over time (Zheng et al., 2024). However, LLMs remain susceptible to catastrophic forgetting, where performance on previous tasks deteriorates after the acquisition of new abilities. (McCloskey and Cohen, 1989).

Despite the extensive methods proposed to mitigate catastrophic forgetting (Wang et al., 2024, 2023b; Zhao et al., 2024), limited studies investigate the intrinsic mechanisms underlying this phenomenon. Kotha et al. (2024) proposed the "task inference" hypothesis, which suggests that fine-tuning biases the model toward utilizing newly acquired capabilities, rather than causing a loss of previously learned abilities. While this hypothesis is validated on synthetic datasets and small transformers, direct empirical evidence from natural language datasets and LLMs is missing. Similarly, Jiang et al. (2024) investigate forgetting in LLMs through the perspectives of instruction-following and task-related knowledge. They highlight that the forgetting stems from a decline in instruction-following capabilities rather than an actual loss of task-related knowledge. Nevertheless, they employ disparate experimental settings—instruction-following for model training versus prefix completion for knowledge probing—which weakens the persuasion of their conclusions.

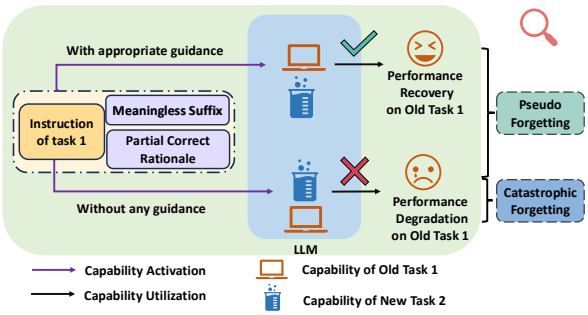


Figure 1: Pseudo forgetting. **1.** The performance degradation on previous tasks stems from instructions failing to properly activate the model's inherent capabilities rather than genuine forgetting of task-relevant abilities. **2.** Performance can be restored through appropriate prompting, demonstrating no actual forgetting occurs.

In this paper, as shown in Figure 1, we argue that the observed performance degradation on previous tasks stems not from a genuine loss of task capabilities, but rather from the instructions' failure to ef-

fectively activate the model's intrinsic abilities—a phenomenon we term "pseudo forgetting". To validate this hypothesis, we conduct probing experiments on LLMs across a range of natural language tasks under instruction-following settings. We find that, given partial rationale as external guidance or augmented with a task-irrelevant instruction suffix, the forgetting model can complete the rationale and reach or even outperform pre-forgetting models, providing strong empirical support for our hypothesis. To investigate the underlying causes of pseudo forgetting, we employ attribution scores to quantitatively analyze the model's reliance on the instructions during rationale generation. Our analysis reveals that the pseudo-forgetting model exhibits significantly reduced reliance on instructions, which prevents the model from effectively utilizing its internal capabilities.

Building on the above insights, we believe that when learning new tasks, replaying data related to previous tasks to strengthen the model's reliance on corresponding instructions offers a simple and effective solution to mitigate pseudo forgetting. However, how to allocate replay data efficiently is limited studied (Wang et al., 2024). Thus, we first introduce the Rationale-Guidance Difficulty (RGD) metric, which measures the model's ability to leverage the correct internal capability under a given instruction. We then propose Rationale-Guidance Difficulty based Replay (RGD-R) to optimize the data utilization in replay-based continual learning algorithms. Specifically, during continual learning, the RGD score for each previous task is dynamically computed and used to determine the ratio of required replay data. Experimental results demonstrate that RGD-R effectively alleviates pseudo forgetting while preserving the model's plasticity.

Our contributions can be summarized as follows:

1. We directly demonstrate the existence of pseudo forgetting in the continual learning of LLMs (Section 2.1), followed by an analysis of the underlying cause(Section 2.2).

2. Building on this insight, we introduce RGD score, which measures the model's ability to leverage the correct intrinsic capabilities under a given instruction (Section 3.1).

3. By adopting RGD, we develop RGD-R, a novel replay-based framework designed to maximize the efficiency of replay data via dynamic data allocation(Section 3.3).

2 Unveiling Pseudo Forgetting : the evidence and cause

Pseudo Forgetting

Pseudo forgetting is a phenomenon where performance degradation on previously learned tasks in continual learning occurs not through the loss of task capabilities, but rather through the diminished effectiveness of original task instructions in activating the model's intact intrinsic capabilities, resulting in incorrect rationales and outputs.

In Section 2.1, we directly demonstrate that models do not genuinely forget task capabilities by restoring their performance on previous tasks via employing two methods to provide appropriate guidance. In Section 2.2, we quantify the model's reliance on instructions during rationale generation, revealing that pseudo forgetting occurs because original instructions fail to activate the model's appropriate intrinsic capabilities.

2.1 Evidence for Pseudo Forgetting

For a forgetting model, two fundamental questions naturally arise:

1. **Q1**: *How does the model perform when **passively** provided with external correct rationale?*

2. **Q2**: *Can changing prompt (eg. adding task-irrelevant prefixes or suffixes) enable the model to generate the correct rationale **actively**?*

A1: With a partially correct rationale guidance, the model can passively recover task performance.

Experiment Setting To address **Q1**, we select the model from the final stage of sequential learning and choose the test set of tasks with a high forgetting rate for this experiment. To offer external correct capability guidance, as shown in Figure 3, the first k words of the ground truth rationale after the <|assistant|> token, where k is the ratio range from 0 to 1 ($k \in [0, 1]$). Notably, providing a small ratio of the correct rationale does not directly reveal task-specific answers, but rather guides the model in shaping the overall direction of its predictions.

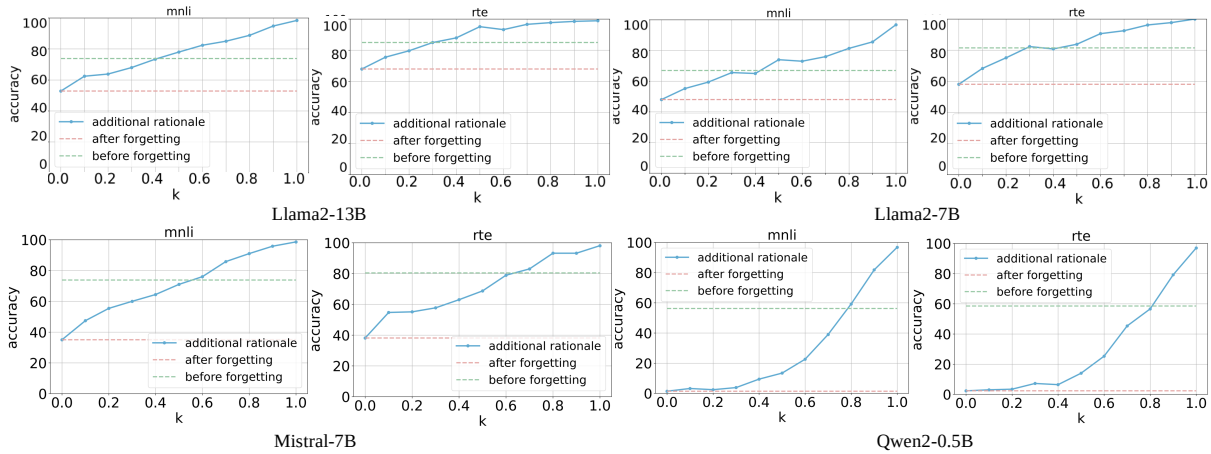


Figure 2: Changes in the model's task performance after forgetting when the first k words of the appropriate rationale are provided. **1.** A forgetting model can regenerate the "forgotten rationale" and gradually recover its "pre-forgetting" task performance when passively guided with partial "appropriate rationale." **2.** the degree of recovery of the task performance is related to the task difficulty and the scale of the model.

<|user|>
Task: What is the logical relationship (contradiction, entailment or neutral) between the "sentence 1" and the "sentence 2"? Choose one from the option.

OPTIONS:
- neutral
- entailment
- contradiction

sentence 1: Case Study Evaluations.
sentence 2: Case Study preparations.

Answer:
<|assistant|>
The sentence 1 'Case Study Evaluations' implies a

Figure 3: Prompt example with additional the first 10% words of the correct rationale guidance ($k = 0.1$). The black parts are the original instruction; The blue parts are the added part of the correct rationale, which does not contain information directly related to the answer.

Results and Analysis The result is illustrated in Figure 2. Firstly, **under the guidance of correct rationale, a forgetting model can recover its task performance to pre-forgetting levels**. Specifically, the performance on different forgotten tasks improves consistently across varying model scales as the value of k increases. Secondly, **the degree of recovery of the model's task performance is related to the task difficulty and the scale of the model**. For instance, in the RTE task, Llama2-13B returns to its pre-forgetting performance level at $k = 0.3$, while the MNLI task requires $k = 0.4$ to achieve the same recovery level. Meanwhile, to restore task performance to pre-forgetting levels steadily, Qwen2-0.5B, Mistral-7B, Llama2-7B, and Llama2-13B require k values of 0.8, 0.6, 0.5, and 0.4, respectively.

Two potential explanations can be proposed:

(1). **Complete catastrophic forgetting**: LLMs require external reasoning guidance to restore performance (even Llama2-13B at $k = 0.4$), suggesting they may simply leverage provided solution components rather than retain problem-solving abilities.

(2). **Capability activation failure**: LLMs' improved performance under minimal guidance indicates preserved capabilities, as critical reasoning steps were self-generated rather than externally provided (as shown in Figure 3, when $k = 0.1$, no significant problem-solving information was explicitly provided).

A2: With the addition of meaningless suffixes, the model can actively recover task performance.

To determine which of these two explanations is correct, we address **Q2**. We employ Greedy Coordinate Gradient (GCG) (Zou et al., 2023) to search a meaningless suffix that helps the original instruction guide the forgetting model toward proper rationale generation actively (Figure 4).

GCG-based Search Given a sequence $x_{1:n}$, the probability of generating a sequence $x_{n+1:n+T}$ can be written as:

$$p(x_{n+1:n+T} \mid x_{1:n}) = \prod_{i=1}^{T} p(x_{n+i} \mid x_{1:n+i-1}) \quad (1)$$

Under the above notation, the loss of generating a target sequence $T = x_{1:N_{target}}$ (eg. partial correct

<|user|>
Task: What is the logical relationship (contradiction, entailment or neutral) between the "sentence 1" and the "sentence 2"? Choose one from the option.

OPTIONS:
- neutral
- entailment
- contradiction

sentence 1: Case Study Evaluations.
sentence 2: Case Study preparations.

Answer: ! involving ! ! dass ! ! ! $\{ ! ! ! ! ! ! ! ! Given ! !
<|assistant|>

Figure 4: Prompt example with *task-irrelevant suffix* searched by Greedy Coordinate Gradient (Zou et al., 2023). The forgetting model outputs *Health and Wellness* due to the influence of the previous task, Yahoo, but correctly outputs *entailment* before forgetting or augmenting with this suffix.

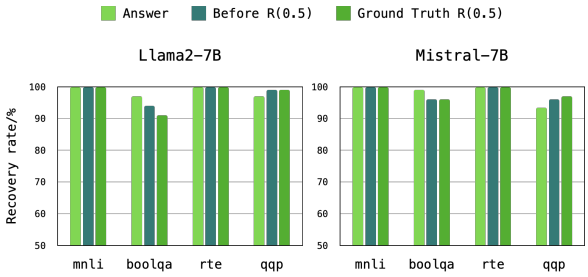


Figure 5: Recovery rate of forgotten tasks. **1.** For each task, we sample 100 forgotten instances. **2.** The labels 'Answer', 'Before R (0.5)', and 'Ground Truth R (0.5)' denote respectively: the ground truth answer, the first 0.5 words of the rationale generated by the model before forgetting and the ground truth, serving as optimization target for GCG. **3.** forgetting sample recovery rates surpass 90% (reaching 100% in specific tasks), indicating the forgetting model preserves previously acquired capabilities.

rationale) given an instruction $I = x_{1:N_{ins}}$ and an initial suffix $S = x_{1:N_{suffix}}$ can be written as

$$\mathcal{L}(S) = -\log p(T \mid [I, S]) \quad (2)$$

To minimize the above loss, GCG (Zou et al., 2023) leverages gradients with respect to the one-hot token indicators to identify promising token replacements. Specifically, for each token position i, in the suffix, the gradient $\nabla \mathcal{L}_{e_i}(S)$ is computed, where e_i is the one-hot vector representing the current token at position i. Then, for each token position, the top-k tokens with the largest negative gradients are identified as candidate replacements. Finally, the candidate replacement that minimizes the loss is selected and applied to the suffix.

Notably, this approach ensures the validity of the experiments: (1) semantically meaningless suffixes devoid of task-specific information, ensuring the generated rationale reflects parametric capabilities; (2) instruction-following setting remains unchanged, aligning the detected capabilities with those learned via instruction fine-tuning, in contrast to the probing experiments in Jiang et al. (2024), which is under prefix completion setting.

Experimental Settings We evaluate models from the final stage of sequential learning. For each task, we sample 100 instances where models exhibit correct predictions before forgetting but fail after forgetting. For GCG, as shown in Table 7, we explore three optimization targets: (1) Answer guidance; (2) Partial ground truth rationale guidance; (3) Partial pre-forgetting rationale guidance. See Appendix B.2 for the detailed implementation.

Results and Analysis As shown in Figure 5, appending task-irrelevant suffixes to original instructions enables forgetting models to actively generate correct rationale, leading to 90% recovery rate across tasks. This provides direct evidence that the model dose not forget previously acquired capabilities. Specifically, the recovery effectiveness may correlate with sample complexity. While Mistral-7B demonstrates complete recovery (100%) on MNLI, its average recovery rate on QQP is 95.44%, with a similar trend observed in Llama2-7B. As detailed in Table 8, the optimal suffix varies across samples, highlighting the dependence of correct rationale generation on the prompt.

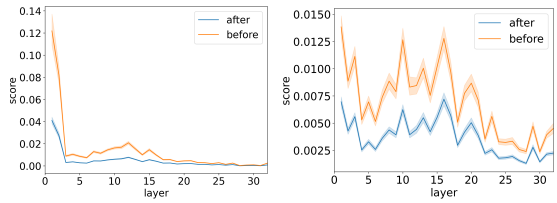
Summary

The results of the two experiments provide direct evidence of pseudo forgetting: the model does not truly forget task-specific capabilities, rather, the original instructions fail to guide the model in leveraging the appropriate abilities to solve the task.

2.2 Cause of Pseudo Forgetting

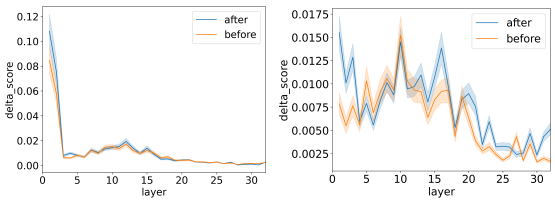
In this section, we investigate the cause of pseudo forgetting to further validate our hypothesis. We demonstrate that the pseudo-forgetting model exhibits a reduced reliance on the original instructions during rationale generation, preventing the model from correctly leveraging its intrinsic capabilities.

Attribution Algorithm We use attribution scores (Li et al., 2024a; Wang et al., 2023a; Dai et al., 2022) to quantify and analyze the dependency between instructions and the generated rationales.



(a) Llama2-7B (b) Mistral-7B

Figure 6: Comparison of instruction dependency scores of pseudo-forgetting model for generating correct and incorrect rationales on MNLI task.



(a) Llama2-7B (b) Mistral-7B

Figure 7: Comparison of relative instruction dependency scores across different states of Llama2-7B and Mistral-7B on MNL task.

Specifically, we can use the Riemann approximation of the integral to calculate the contribution of a neuron ω to the model's output $F(\cdot)$, with m approximation steps:

$$\text{Attr}(\omega) = \omega \circ \int_0^1 \frac{\partial F(\alpha\omega)}{\partial \omega} d\alpha \approx \frac{\omega}{m} \sum_{k=1}^{m} \frac{\partial F\left(\frac{k}{m}\omega\right)}{\partial \omega} \quad (3)$$

Since the self-attention layers learn strong instruction-following patterns (Wu et al., 2024), we can compute the dependency between the instruction $I = x_1 : x_{N_{ins}}$ and the given rationale $R = x_1 : x_{N_{rationale}}$ based on the attention layers:

$$Q_{IR}^{(l)} = \frac{1}{|N|} \sum_{(i,j) \in D_{IR}} \text{Attr}(A_{i,j}^{(l)}) \quad (4)$$

$$D_{IR} = \{(i,j) | x_i \in I, x_j \in R\} \quad (5)$$

In this notation, $\text{Attr}(A_{i,j}^{(l)})$ represents the dependence intensity from the i-th token to the j-th token in the l-th self-attention layer, calculated by summing the absolute attribution scores across all heads. $|N|$ denotes the total number of rationale steps. More implementation details are provided in the Appendix B.3.

Experimental Settings We use M_{b-f} and M_{a-f} to denote the model trained on the old task and continually trained on the final task, corresponding to the stages of before and after pseudo forgetting. The probing dataset is the same as that used in Section 2.1. Each sample can be denoted as $(I, R_{b-f}, R_{a-f}, R_g, A_{b-f}, A_{a-f}, A_g)$, where I represents the instruction, R_{b-f}, R_{a-f}, R_g represent the rationale generated by M_{b-f}, M_{a-f}, and Llama3.1-70B-Instruct (as the ground truth), respectively. A_{b-f}, A_{a-f}, A_g represent the corresponding predicted answers.

Experiment 1 Firstly, we investigate the differences in the pseudo-forgetting model's (M_{a-f}) instruction dependency when generating incorrect (R_{a-f}) versus correct (R_{a-f}) rationale.

As shown in Figure 6, we can conclude that **the pseudo-forgetting model generates incorrect rationales primarily due to the reduced instruction dependency**. Specifically, for M_{a-f}, the instruction dependency when generating incorrect rationales (blue line) is generally lower than that when generating correct rationales (orange line). The difference is noticeable in shallow layers, aligning with the findings in Wu et al. (2024) that shallow layers learn more and stronger instruction-following patterns.

Experiment 2 Secondly, to confirm that the reduced instruction dependency is indeed caused by pseudo forgetting, we examine the impact of different models (M_{b-f} vs M_{a-f}). Specifically, we compare the relative instruction dependency scores when different models generate rationales:

$$\Delta_{\text{Attr}(R_{gen}|R_g)} = |Q_{IR_{gen}}^{(l)} - Q_{IR_g}^{(l)}| \quad (6)$$

where R_{gen} is R_{a-f} (R_{b-f}) if we calculate Equation (6) on M_{a-f} (R_{b-f}). This approach ensures that the only variable in the experiment is the occurrence of pseudo forgetting.

As shown in Figure 7, the discrepancy between R_g and R_{a-f} on M_{a-f} (blue line) is larger compared to the difference between R_g and R_{b-f} on M_{b-f} (orange line)[1]. This finding further supports our hypothesis that **a key factor contributing to pseudo forgetting is the model's reduced reliance on the original instruction during rationale generation**.

[1]While certain layers display differences or larger "before" delta scores compared to "after" scores, analyzing these observations is outside the scope of this work.

3 Addressing Pseudo Forgetting: Rationale-Guidance Difficulty based Replay

Based on these findings, we argue that replay-based algorithms, which incorporate a small portion of data from previous tasks during continual learning, can effectively reinforce the model's dependency on corresponding instructions, thereby offering a simple yet effective solution to pseudo forgetting. However, how to allocate the replay data ratio for each task remains underexplored (Wang et al., 2024). Thus, in Section 3.1, we introduce the Rationale-Guidance Difficulty (RGD) metric to measure the impact of pseudo forgetting on the model. Then, in Section 3.3, we propose Rationale-Guidance Difficulty based Replay (RGD-R), which leverages RGD to dynamically determine the replay data proportion for each task, optimizing replay data utilization during continual learning.

3.1 Rationale-Guidance Difficulty

We first introduce the Rationale-Guidance Difficulty (RGD) metric, which measures the difficulty for the model to correctly utilize its internal capabilities in generating appropriate rationale under a given instruction. For a data triplet (I, R_g, A_g), the RGD score [2] is calculated as follows:

$$\mathsf{RGD}(I, R_g, A_g) = \frac{\mathsf{PPL}_{a-f}(R_g|I)}{\mathsf{PPL}_{b-f}(R_g)} \tag{7}$$

where I, R_g, and A_g denote the prompt, the ground truth rationale, and the ground truth answer, respectively. $\mathsf{PPL}_{b-f}(R_g)$ represents the difficulty for the model with normal access to its capabilities to generate the correct rationale, and $\mathsf{PPL}_{a-f}(R_g|I)$ denotes the difficulty for the pseudo-forgetting model to generate the same rationale given prompt I. A higher RGD score signifies greater difficulty for a prompt in guiding the model to generate the rationale, and vice versa.

$$\mathsf{RGD}_D = \frac{1}{|D|}\sum_i \mathsf{RGD}(I, R_g, A_g)_i \tag{8}$$

where $(I, R_g, A_g)_i$ is the i-th sample in dataset D, and $|D|$ is the total number of samples.

3.2 Theoretical Analysis

Here, we give a simple proof that under a reasonable assumption, the RGD score can measure the difficulty of the capability activation process. First, Wu et al. (2024) finds that the underlying mechanism of instruction following likely involves model θ first recognizing instruction i, then utilizing the activated capabilities $c_1, \ldots, c_n$ to generate rationale r. We can formalize this process as:

$$P_\theta(r|i) = \sum_n p(r \mid c_n) \cdot p(c_n \mid i) \tag{9}$$

Assumption. *Under normal circumstances, each capability c can only be activated by task-specific instructions i, which subsequently supports the generation of the corresponding rationale r. The capabilities of tasks across different domains are independent from one another.*

$$\forall m \neq n, \quad p(r \mid c_n) \cdot p(c_m \mid i) = 0 \tag{10}$$

We can formalize the probability of activating the correct task capability c^* given instruction i as:

$$P_\theta(c^* \mid i) = p(c_1, \ldots, c_m \mid i) = \sum_m p(c_m \mid i) \tag{11}$$

The correct rationale generation process is:

$$P_\theta(r^*) = p(r^* \mid c_1, \ldots, c_n) = \sum_m p(r^* \mid c_n) \tag{12}$$

Based on assumption 3.2, we can rewrite Equation 9 as:

$$P_\theta(r^*|i) = \Big(\sum_n p(r^* \mid c_n)\Big) \cdot \Big(\sum_m p(c_m \mid i)\Big) \tag{13}$$

Hence, the following equation holds:

$$P_\theta(c^* \mid i) = \frac{P_\theta(r^*|i)}{P_\theta(r^*)} \tag{14}$$

Consequently, the RGD score can approximate the difficulty of a given instruction in activating the model's corresponding capability.

3.3 RGD-based Replay framework

To optimize the data utilization in replay-based methods, we propose the Rationale-Guidance Difficulty-based Replay (RGD-R) framework. During continual learning, RGD-R dynamically determines the required replay data ratio for each previous task based on the RGD score calculated via Equation 8. Specifically, when training the model on the i-th task, the replay data ratio for the j-th previous task can be calculated as:

$$\alpha_j = \frac{\mathsf{RGD}_{D_j}}{\sum_{k=1}^{i-1} \mathsf{RGD}_{D_k}}, \quad j \in [1, i-1] \tag{15}$$

where $\sum_{j=1}^{i-1} \alpha_j = 1$, and RGD_{D_j} represents the RGD score of the j-th previous task. Thus, the amount of replay data allocated to this task is $\alpha_j \cdot N$, where N represents the total amount of replay data.

[2] This metric is calculated similarly to Instruction Following Difficulty score (Li et al., 2024c), which is mainly used for data selection (Li et al., 2024b,c)

3.4 Experiments

3.4.1 Experiment Setting

Datasets Following Razdaibiedina et al. (2023a) and Wang et al. (2023c), we conduct experiments on Long Sequence Benchmark, with train/validation/test splits of 1000/500/500 samples respectively. See Appendix A for more details.

Metrics Following prior works (Zhao et al., 2024; Zhang et al., 2023b) Let $a_{i,j}$ be the testing performance on the j-th task after training on i-th task, the metrics for evaluating are: (1) **Final Average Performance (FAP)** is the average performance of all tasks after the final task t_T is learned, i.e., $\texttt{FAP}_T = \frac{1}{T}\sum_{t=1}^{T} a_{T,t}$; (2) **Forgetting Rate (F.Ra)** measures how much knowledge has been forgotten across the first $T-1$ tasks, i.e., $\text{F}_T = \frac{1}{T-1}\sum_{t=1}^{T-1}(\max_{k=i}^{T-1} a_{k,t} - a_{T,t})$; (3) **Backward Transfer (BWT)** measures the impact that continually learning on subsequent tasks has on previous tasks, i.e., $\texttt{BWT}_T = \frac{1}{T-1}\sum_{t=1}^{T-1}(a_{T,t} - a_{t,t})$. (4) **Forward Transfer (FWT)** measures how much the model can help to generalize and learn the new task, i.e., $FWT = \frac{1}{T}\sum_{t=2}^{T} a_{t-1,t}$. Better scores on FAP, F.Ra, and BWT indicate improved model stability, while a better FWT score reflects enhanced model plasticity.

Baselines To validate the effectiveness of RGD in measuring pseudo forgetting and RGD-R in mitigating this phenomenon, we conduct comparative experiments across the following baselines focusing on replay data allocation, where samples for each task are randomly selected from the training set: (1) **Sequential Training (SEQ)** refers to learning new capabilities without replay data. (2) **Equal Allocation (EA)** replays the same amount of data for each previous task. See Appendix B.1 for more details.

3.4.2 Main Results

LLMs exhibit inherent resistance to pseudo forgetting, which improves with larger model sizes. Larger models show lower forgetting rates, such as F.Ra of Llama2-13B and Qwen2-0.5B with SEQ are 13.54 and 53.18, respectively.

The equal allocation method significantly alleviates pseudo forgetting. Compared to SEQ, EA improves the final performance (FAP) of Qwen2-0.5B, Mistral-7B, and Llama2-13B by 43.40, 20.67, and 8.60, respectively, while reducing the forgetting rate (F.Ra) by 49.54, 22.6, and 9.86. These results support our hypothesis that LLMs do not truly forget the previously learned capabilities.

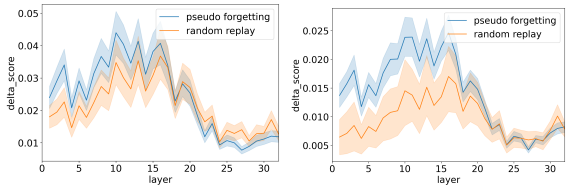
Method	FAP↑	F.Ra↓	BWT↑	FWT↑
Qwen2-0.5B				
SEQ	20.73	53.18	-53.04	21.46
EA	64.13	5.43	-4.90	33.34
RGD-R	**65.99**	**3.64**	**-3.29**	**31.87**
Mistral-7B				
SEQ	51.48	30.19	-29.97	47.91
EA	72.15	7.59	-6.96	**51.17**
RGD-R	**74.91**	**4.37**	**-3.92**	50.77
Llama2-7B				
SEQ	62.79	17.87	-17.85	43.95
EA	76.10	3.52	-2.49	50.91
RGD-R	**77.03**	**2.65**	**-1.25**	**51.06**
Llama2-13B				
SEQ	68.38	13.54	-13.2	51.69
EA	76.98	4.73	-3.70	56.92
RGD-R	**78.25**	**3.68**	**-2.29**	**57.83**

Table 1: Performance of different models on Long Sequence Benchmark. The decoding strategy is greedy search. RGD-R effectively alleviates model forgetting and maintains better model plasticity simultaneously.

RGD-R further alleviates pseudo forgetting and ensures the model plasticity simultaneously. Compared to EA, RGD-R demonstrates superior effectiveness in mitigating pseudo forgetting (FAP, F.Ra, BWT) and promoting asynchronous knowledge transfer (FWT) across different models. This highlights the efficacy of the RGD score in measuring the impact of pseudo forgetting and confirms that RGD-R successfully optimizes the utilization of replay data in replay-based continual learning algorithms, leading to better overall model performance.

3.4.3 Analysis

Data Replay Restores Instruction dependence To demonstrate that the replay-based method indeed enhances the instruction dependence, we repeat the attribution experiment in Section 2.2. Specifically, we compare the relative instruction dependency scores between the pseudo-forgetting model trained via SEQ and the model trained via EA data replay. As shown in Figure 8, the model trained via data replay (orange line) exhibits a smaller overall difference in instruction dependence when generating rationales compared to the pseudo-forgetting model (blue lines). This suggests that the replay-based method improves the model's reliance on original instructions, thereby alleviating pseudo forgetting.



(a) MNLI (b) QQP

Figure 8: Comparison of relative instruction dependency scores across different states of Mistral-7B on MNLI and QQP tasks. **1.** 'pseudo forgetting' and 'random replay' represent Mistral-7B exhibiting pseudo forgetting and Mistral-7B after capability recovery through random data replay, respectively. **2.** The replay-based method leads to lower relative instruction dependency scores, indicating that it helps the model rely more on instructions during rationale generation.

Rationale	MNLI	BOOLQA	RTE
R_{a-f}	0.2756	0.2962	0.2538
$R_{Paraphrase}$	0.6641	0.6793	0.6554
R_{GCG}	0.2871	0.3134	0.2856
R_g	0.4103	0.4719	0.4038
R_{Replay}	**0.4391**	**0.4931**	**0.4359**

Table 2: Comparison of ROUGE-L scores between rationales ($R_{(\cdot)}$) generated by different methods and those (R_{b-f}) from the model before pseudo-forgetting. **1.** $R_{Paraphrase}$ is the paraphrased rationale generated by GPT-3.5 based on R_{b-f}. R_{GCG} and R_{Replay} are the rationales generated after mitigating pseudo forgetting with the GCG and data replay methods, respectively.

Data Replay Enables Better Semantic Recovery in Rationales We compare the semantic similarity between rationales generated by different methods ($R_{(\cdot)}$) and those generated by the pre-pseudo-forgetting model (R_{a-f}). As shown in Table 2, the replay-based method achieves higher semantic similarity compared to GCG, and surpasses the ground truth rationales. This indicates that replay-based methods are more effective in stimulating the model's previously learned task capabilities. In contrast, based on GCG, the pseudo-forgetting model still tends to generate tokens related to the new task (Gu and Feng, 2020). While adding a semantic constraint to GCG helps alleviate this issue, our preliminary experiments show that it makes the search process harder and less efficient.

4 Related Work

Mechanism of catastrophic forgetting While many continual learning algorithms are proposed, a substantial gap persists in understanding the mechanism of catastrophic forgetting. Kotha et al. (2024) hypothesize that models first perform "task inference" before applying the relevant capability, and fine-tuning biases this inference towards tasks aligned with the fine-tuning distribution, thereby suppressing performance on other prior capabilities. Jiang et al. (2024) believe that forgetting is primarily due to the reduced instruction-following capability, rather than a loss of task-related knowledge. Unlike our work, the above studies do not provide direct and effective evidence of pseudo forgetting on LLMs and natural language datasets.

Traditional methods in continual learning (1) *Regularization-based* methods constrain the features learned from previous tasks (Zhang et al., 2023a; Huang et al., 2021) or penalize changes to weights critical for those tasks (Zhou and Cao, 2021; Wang et al., 2023b), ensuring that new learning minimally interferes with prior capability thus maintaining performance on earlier tasks. (2) *Architecture-based* methods aim to reduce the interference by either increasing the model's capacity (Zhao et al., 2024) or isolating the existing weights (Hu et al., 2024). (3) *Replay-based* methods retain a small subset of prior training examples or pseudo data and revisit them when a new task is introduced (Guo et al., 2024; Huang et al., 2024; Qin and Joty, 2022). InsCL (Wang et al., 2024) allocates replay data based on the similarity of task instructions. In this paper, we introduce RGD-R, which dynamically allocates replay data based on the model's susceptibility to pseudo forgetting, capturing more model-relevant characteristics to help the model maintain both stability and plasticity.

5 Conclusion

In this study, we directly demonstrate the phenomenon of "pseudo forgetting" in LLMs during continual learning. We show that the performance degradation on previous tasks does not stem from the loss of corresponding capabilities, but rather from reduced instruction dependence during rationale generation. We introduce the RGD score to quantify the extent of the model's susceptibility to pseudo forgetting, which is then used to dynamically allocate the replay ratio for each previous task to optimize replay data utilization in our proposed RGD-R framework. Experimental results confirm the effectiveness of RGD-R in addressing pseudo forgetting and preserving model plasticity.

Limitations and Future Works

While this paper analyzes and addresses pseudo forgetting during continual learning in LLMs, several limitations warrant further discussion. First, we do not conduct an in-depth analysis of the specific process behind pseudo forgetting. For instance, at what point during the learning of new tasks does the model begin to show reduced dependence on the instructions from previously learned tasks? What are the underlying factors driving this decline? Second, the relationship between pseudo forgetting and specific tasks or domains remains unexplored. For example, as noted by Li et al. (2024d), domain generalization in summarization tasks correlates with words distribution, raising the question of whether pseudo forgetting exhibits similar characteristics. Additionally, we propose that measuring pseudo-forgetting is likely a multi-dimensional problem, and our proposed RGD score represents just one possible metric. The development of more comprehensive evaluation metrics for this phenomenon requires additional research. Finally, our findings indicate that LLMs do not forget previously acquired capabilities, and Dai et al. (2022) suggest that these capabilities are stored parametrically within the model. Consequently, to optimize continual learning algorithms, we suggest that future works could benefit from combining replay-based and parameter-based approaches, with a greater emphasis on enhancing asynchronous knowledge transfer capabilities—an underexplored aspect in current research (Zhang et al., 2023b).

6 Ethics Statement

This work focuses on analyzing and addressing pseudo forgetting in large language models during continual learning, and as such, does not introduce additional ethical risks beyond those inherent to standard NLP research. The potential risks primarily stem from two aspects: First, our experiments utilize large language models trained on vast amounts of internet text data, which may contain societal biases. However, since our research focuses on analyzing model capabilities rather than deploying systems, the risk of propagating harmful biases is minimal. Second, while our findings about model capabilities and instruction dependence could potentially be misused to manipulate model outputs, our work specifically aims to improve model reliability and performance stability, ultimately contributing to more robust and dependable AI systems. Throughout our experiments, we used standard benchmarks and publicly available datasets to ensure reproducibility and transparency. Our methods and findings are intended to advance the scientific understanding of continual learning in language models while adhering to established ethical guidelines in NLP research.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Shuhao Gu and Yang Feng. 2020. Investigating catastrophic forgetting during continual training for neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4315–4326, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jiafeng Guo, Changjiang Zhou, Ruqing Zhang, Jiangui Chen, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. Corpusbrain++: A continual generative pre-training framework for knowledge-intensive language tasks. *CoRR*, abs/2402.16767.

Yusong Hu, De Cheng, Dingwen Zhang, Nannan Wang, Tongliang Liu, and Xinbo Gao. 2024. Task-aware orthogonal sparse network for exploring shared knowledge in continual learning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1416–1428. Association for Computational Linguistics.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 2021. Continual learning for text classification with information disentanglement based regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2736–2746. Association for Computational Linguistics.

Gangwei Jiang, Caigao Jiang, Zhaoyi Li, Siqiao Xue, Jun Zhou, Linqi Song, Defu Lian, and Ying Wei. 2024. Interpretable catastrophic forgetting of large language model fine-tuning via instruction vector. *CoRR*, abs/2406.12227.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. 2024. Understanding catastrophic forgetting in language models via implicit inference. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jiachun Li, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Daojian Zeng, Kang Liu, and Jun Zhao. 2024a. Focus on your question! interpreting and mitigating toxic cot problems in commonsense reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9206–9230. Association for Computational Linguistics.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024b. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 14255–14273. Association for Computational Linguistics.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024c. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 7602–7635. Association for Computational Linguistics.

Yinghao Li, Siyu Miao, Heyan Huang, and Yang Gao. 2024d. Word matters: What influences domain adaptation in summarization? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 13236–13249. Association for Computational Linguistics.

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

Chengwei Qin and Shafiq R. Joty. 2022. LFPT5: A unified framework for lifelong few-shot language learning based on prompt tuning of T5. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023a. Progressive prompts: Continual learning for language models. In *International Conference on Learning Representations*.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023b. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023a. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore. Association for Computational Linguistics.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023b. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10658–10671. Association for Computational Linguistics.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023c. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, Singapore. Association for Computational Linguistics.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024. Inscl: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 663–677. Association for Computational Linguistics.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023d. How far can camels go? exploring the state of instruction tuning on open resources. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Xuansheng Wu, Wenlin Yao, Jianshu Chen, Xiaoman Pan, Xiaoyang Wang, Ninghao Liu, and Dong Yu. 2024. From language modeling to instruction following: Understanding the behavior shift in LLMs after instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2341–2369, Mexico City, Mexico. Association for Computational Linguistics.

Yizhe Yang, Huashan Sun, Jiawei Li, Runheng Liu, Yinghao Li, Yuhang Liu, Heyan Huang, and Yang Gao. 2023. Mindllm: Pre-training lightweight large language model from scratch, evaluations and domain applications. *Preprint*, arXiv:2310.15777.

Duzhen Zhang, Wei Cong, Jiahua Dong, Yahan Yu, Xiuyi Chen, Yonggang Zhang, and Zhen Fang. 2023a. Continual named entity recognition without catastrophic forgetting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8186–8197. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Zihan Zhang, Meng Fang, Ling Chen, and Mohammad-Reza Namazi-Rad. 2023b. CITB: A benchmark for continual instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9443–9455, Singapore. Association for Computational Linguistics.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. SAPT: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11641–11661. Association for Computational Linguistics.

Junhao Zheng, Shengjie Qiu, Chengming Shi, and Qianli Ma. 2024. Towards lifelong learning of large language models: A survey. *CoRR*, abs/2406.06391.

Fan Zhou and Chengtai Cao. 2021. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4714–4722. AAAI Press.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

A Dataset Details

A.1 Datasets

Long Sequence Benchmark The Long Sequence Benchmark (Razdaibiedina et al., 2023b) comprises 15 tasks from CL benchmark (Zhang et al., 2015), GLUE benchmark (Wang et al., 2019b), and SuperGLUE benchmark (Wang et al., 2019a), as detailed in Table 3.

Dataset	Source	Task	Domain	Metric
1. Yelp	CL Benchmark	sentiment analysis	Yelp reviews	accuracy
2. Amazon	CL Benchmark	sentiment analysis	Amazon reviews	accuracy
3. DBpedia	CL Benchmark	topic classification	Wikipedia	accuracy
4. Yahoo	CL Benchmark	topic classification	Yahoo Q&A	accuracy
5. AG News	CL Benchmark	topic classification	news	accuracy
6. MNLI	GLUE	natural language inference	various	accuracy
7. QQP	GLUE	paragraph detection	Quora	accuracy
8. RTE	GLUE	natural language inference	news, Wikipedia	accuracy
9. SST-2	GLUE	sentiment analysis	movie reviews	accuracy
10. WiC	SuperGLUE	word sense disambiguation	lexical databases	accuracy
11. CB	SuperGLUE	natural language inference	various	accuracy
12. COPA	SuperGLUE	question and answering	blogs, encyclopedia	accuracy
13. BoolQA	SuperGLUE	boolean question and answering	Wikipedia	accuracy
14. MultiRC	SuperGLUE	question and answering	various	accuracy
15. IMDB	SuperGLUE	sentiment analysis	movie reviews	accuracy

Table 3: The details of 15 classification datasets in the Long Sequence Benchmark (Razdaibiedina et al., 2023b).

A.2 Task Sequence Orders

Following previous works (Zhao et al., 2024; Razdaibiedina et al., 2023b), we conduct experiments using two different training orders, as shown in Table 4.

Order	Task Sequence
1	mnli → cb → wic → copa → qqp → boolqa → rte → imdb → yelp → amazon → sst-2 → dbpedia → ag → multirc → yahoo
2	yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yahoo → multirc → boolqa → wic

Table 4: Tow different orders of task sequences used for our experiments correspond to the Long Sequence Benchmark.

A.3 Data Construction and Ground Truth Rationales Generation

The raw sample consists of an instruction I, an input I_{input}, and an answer A. We adopted the instruction conversion templates proposed by Wang et al. (2023d) to integrate inputs into instructions ($[I, I_{input}] \rightarrow I$). To explicitly probing the model's acquired capabilities, we employed Llama3.1-70B-Instruct [3] to generate a rationale R for each sample. The final data samples were structured as triples (I, R_g, A_g). Specifically, we use the prompt shown in Table 5 to ensure that A_g would not appear directly within R_g, or would only appear at the end of R_g. This approach prevent the occurrence of A_g being provided via partial rationale guidance in experiments in Section 2.1, thereby ensuring the validity of our experimental results.

[3] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

<|begin_of_text|><|start_header_id|>system<|end_header_id|>
{default system prompt}
<|eot_id|><|start_header_id|>user<|end_header_id|>
Instruction:
We have a question and an answer provided below. Your task is to generate a rationale that explains the reasoning behind the given answer. The rationale should be comprehensive, logical, and clearly support why the answer is appropriate for the question.
QA Pair:
Original question:
{Instruction}
Original answer:
{Answer}
Guidelines:
1. Provide a detailed rationale for the given answer.
2. Ensure that the rationale is clear, logical, and free of any ambiguity.
Format:
Please generate the following JSON formatted output and nothing else:<|eot_id|><|start_header_id|>assistant<|end_header_id|>
{"answer": "{Answer}", "rationale": "The correct answer is {Answer}.
The rationale behind this answer is as follows:

Table 5: The prompt for ground truth rationale generation

B Experimental Implementation Details

B.1 Model Training

We perform continual learning training using the LoRa algorithm on the 7B and 13B models. Specifically, the LoRA hyperparameters are set as follows: $\texttt{lora\_rank} = 8$, $\texttt{lora\_alpha} = 16$, and $\texttt{lora\_dropout} = 0.1$, with LoRA applied across all modules. For the Qwen2-0.5B model, we directly apply full fine-tuning. The detailed parameter settings are presented in Table 6:

Model Size	Optimizer	Lr Scheduler	Learning Rate	Batch Size	Epochs
$\geq 7B$	AdamW	Warmup=0.03 Decay="cosine"	5e-4	32	6
$\leq 7B$	AdamW	Warmup=0.03 Decay="cosine"	5e-5	64	3

Table 6: Training details of continual learning

B.2 GCG Implementation

In Section 2.1, we employ GCG (Zou et al., 2023) to search for the suffix corresponding to each forgotten sample, which enables the original instruction to guide the pseudo-forgetting model in generating appropriate rationale and restoring performance on previous tasks. Specifically, as shown in Table 7, we utilize three optimization objectives to facilitate the search process. The termination conditions are set as: (1) correct model response to the original instruction, or (2) reaching the maximum iteration count of 500. We configure the hyperparameters with $\texttt{top} - k = 256$ and $\texttt{batch size} = 256$. The initial suffix is set to "! ".

Target T	**Example** ($k = 0.2$)
Answer	The answer is: `{ground truth answer}`. The reasons are as follows:
Partial R_g	1. To establish the logical relationship between the two sentences, we must analyze the meaning and implications of each. 2. Sentence 1 states that the presence of a smart doctor who gave a tip through
Partial R_{b-f}	1. Sentence 1 states that there was a smart doctor who gave them a tip through the Coroner,which implies the presence and involvement of a doctor in the situation.

Table 7: Optimization targets used by GCG on MNLI task in Experiment 2.1. **1.** R_g and R_{b-f} represent the ground truth rationale and the rationale generated by the pre-forgetting model, respectively. **2.** The rationale shown here corresponds to the first 20% of the sequence; in the actual experiment, we used the first 50%.

B.3 Attribution Implementation

In Section 2.2, we quantify the model's dependency on the given instruction during rationale generation using an attribution algorithm (Li et al., 2024a; Wang et al., 2023a; Dai et al., 2022). Specifically, in Equation (3), $F(\cdot)$represents the language modeling loss, and $m = 20$. Each sentence in the rationale is treated as a separate reasoning step, allowing us to compute the total number of inference steps, $|N|$, as described in Equation (4).

C Case study

Model	**Task**	**Partial suffixes**
Mistral-7B	BoolQA	`! ! Sounds striking ! ! ! ! ! Bo ..## !phony provisions !="#`
Mistral-7B	BoolQA	`And ! ! ! ! ! doesn ! mentioned ! !However ! ! ! Shadow ! !`
Mistral-7B	MNLI	`! ! ! ! ! ! ! ! ! the ! ! Fifth ! ! ! ! ! ! !`
Mistral-7B	MNLI	`! ! Cons ! > nation ! April ! G ! Pub Final ! ! ! ! ! ! !`
Qwen2-0.5B	MNLI	`!HolAndHashCode ! ErrorResponse-not Donovan unpublished`

Table 8: Examples of instruction suffixes discovered by GCG. Due to length constraints, only the initial portions of the suffixes are shown.