TRACE: A COMPREHENSIVE BENCHMARK FOR CONTINUAL LEARNING IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Aligned large language models (LLMs) demonstrate exceptional capabilities in task-solving, following instructions, and ensuring safety. However, the continual learning aspect of these aligned LLMs has been largely overlooked. Existing continual learning benchmarks lack sufficient challenge for leading aligned LLMs, owing to both their simplicity and the models' potential exposure during instruction tuning. In this paper, we introduce TRACE, a novel benchmark designed to evaluate continual learning in LLMs. TRACE comprises eight challenging tasks from the scope of domain-specific tasks, multilingual capabilities, code generation, and mathematical reasoning. We have conducted systematic analysis experiments on TRACE using six different aligned models, ranging in size from 7B to 70B. Our experiments show that after training on TRACE, aligned LLMs exhibit significant declines in both general ability and instruction-following capabilities. For example, the accuracy of llama2-chat 13B on the gsm8k dataset declined precipitously from 43.14% to 2.12% after training on our datasets. This highlights the challenge of finding a suitable tradeoff between achieving performance on specific tasks while preserving the original prowess of LLMs. Motivated by empirical findings, we introduce the Reasoning-augmented Continual Learning (RCL) approach. RCL integrates task-specific cues with meta-rationales, effectively reducing catastrophic forgetting in LLMs while expediting convergence on novel tasks.

1 INTRODUCTION

Large Language Models (LLMs) (OpenAI, 2023; Touvron et al., 2023) have revolutionized natural language processing through a two-step process: initial pretraining on extensive corpora, followed by fine-tuning on human-generated instructions and preference data, aligning them with human language and intentions. Aligned LLMs have showcased impressive capabilities and ensured safer responses. However, as the demands for language models grow, there's a pressing need to enhance their abilities in areas such as domain-specific knowledge (Wu et al., 2023; Li et al., 2023), multilingual proficiency (Huang et al., 2023), complex task-solving (Surameery & Shakor, 2023), and tool usage (Qin et al., 2023). Yet, retraining and realigning them from scratch to meet these demands is impractical due to prohibitive training costs and the challenge of acquiring high-quality data. Therefore, incrementally training existing Aligned LLMs through continual learning (CL (Wang et al., 2023a)) is crucial. However, when dealing with tasks in sequential manners, it is challenging to retain the performance of previous tasks, which is known as ”catastrophic forgetting” (McCloskey & Cohen, 1989). Therefore, we prompt the pressing question: *To what degree do Aligned LLMs exhibit catastrophic forgetting when subjected to incremental training?*

Existing continual learning benchmarks (Zhang et al., 2015; Scialom et al., 2022; Razdaibiedina et al., 2023) are not suitable for evaluating the state-of-the-art LLMs. Firstly, many of these benchmarks predominantly consist of simplistic natural language understanding datasets. These tasks, due to their inherent simplicity, fail to challenge the capabilities of large-scale models adequately. Secondly, prior benchmarks have primarily focused on metrics that assess the performance of the models on target sequential tasks. Yet, for aligned models, aspects like generalization to new tasks, the ability to follow human instructions, and safety preservation are of paramount importance. Regrettably, these dimensions have not been extensively studied or incorporated into assessments.

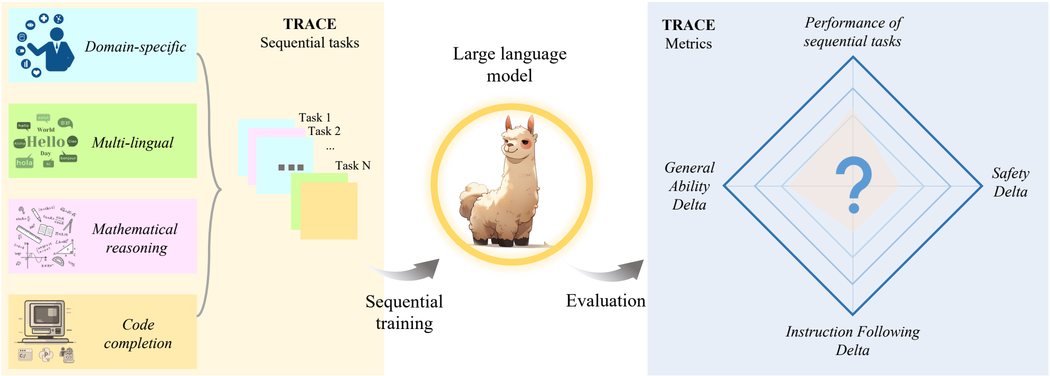


Figure 1: An overview of TRACE benchmark. TRACE consists of two main components: 1) A selection of eight datasets constituting a tailored set of tasks for continual learning, covering challenges in domain-specific tasks, multilingual capabilities, code generation, and mathematical reasoning. 2) A post-training evaluation of LLM capabilities. In addition to traditional continual learning metrics, we introduce General Ability Delta, Instruction Following Delta, and Safety Delta to evaluate shifts in LLM's inherent abilities.

In this paper, we present **TRACE**, a continual learning benchmark designed for aligned Large Language Models. Our benchmark consists of eight distinct datasets spanning challenging tasks including domain-specific tasks, multilingual capabilities, code generation, and mathematical reasoning. To ensure task balance, we sample 5,000 instances for each task, and for classification tasks, we ensure an equal number of samples across all classes. Additionally, all datasets have been standardized into a unified format, simplifying the evaluation process. To evaluate continual learning in aligned LLMs, we introduce three metrics: ”General Ability Delta,” ”Instruction Following Delta,” and ”Safety Delta” to assess models' forgetfulness in such scenarios.

We conduct a comprehensive evaluation of 6 aligned LLMs on TRACE. Evaluation results reveal several key findings: 1) Nearly all models exhibit a significant decline in general abilities after training on TRACE, especially in math and reasoning. For instance, the accuracy of llama2-chat 13B on the gsm8k dataset dropped from 43.14% to merely 2.12%. 2) Catastrophic forgetting remains a substantial issue for LLMs and does not diminish with the model size increase. Llama2-chat 70B also shows significant forgetting (with -17.5% backward transfer) on previous tasks. 3) Full-parameter training, compared to LoRA training, more easily fits the target tasks, but it also leads to a more pronounced decline in general abilities. 4) LLMs' instruction-following capabilities also suffer a significant reduction after continual learning.

Through experimentation, we observed that tasks augmented with reasoning paths are notably effective in preserving certain capabilities of LLMs, preventing them from substantial declines. Such findings lead us to ponder on leveraging a model's inherent strengths for rapid transfer on new tasks, rather than starting the learning curve from scratch. This motivation birthed our novel training strategy: Reasoning-augmented Continual Learning (RCL). RCL prompts the model to generate task analyses and rationales during training. As our results indicate, this approach not only boosts performance on target tasks but also significantly upholds the inherent strengths of LLMs.

2 Related Work

2.1 Continual Learning

Continual learning (Wang et al., 2023a) aims to develop learning algorithms that can accumulate knowledge on non-stationary data. Existing works can be broadly categorized into rehearsal-based, regularization-based, and architecture-based approaches. Rehearsal-based approaches (Lopez-Paz & Ranzato, 2017; de Masson D'Autume et al., 2019) leverage a memory buffer that stores examples from previous tasks, training the model jointly with the current task. Regularization-based approaches (Kirkpatrick et al., 2017; Smith et al., 2023; Wang et al., 2023b) incorporate additional terms into the loss function to penalize changes in crucial weights. Architecture-based approaches (Wang et al., 2023c; Razdaibiedina et al., 2023) focus on dynamically expanding model capacity or isolating existing model weights to mitigate interference between new and old tasks.

However, many continual learning methods face significant challenges when applied to LLMs. For example, for regularization-based approaches, EWC (Kirkpatrick et al., 2017) requires storing parameters from past tasks while GEM (Lopez-Paz & Ranzato, 2017) needs to project every sample to its past loss gradient vectors. OGD (Farajtabar et al., 2020) also needs to store all the gradients of past samples. These methods not only pose challenges to computation and storage capabilities but are often practically infeasible for large models due to severe hardware limitations. Architecture-based methods (such as L2P (Wang et al., 2022b) and PP (Razdaibiedina et al., 2023)), often based on prompt tuning (Lester et al., 2021), but their requirement for task-ID can be impractical in many real-world scenarios. Replay-based methods (such as ER (Lopez-Paz & Ranzato, 2017) and LFPT5 (Qin & Joty, 2022)) raise privacy concerns for the storage and replay of data from previous tasks. Specifically, LFPT5 (Qin & Joty, 2022) employs synthetic data replay generated from task-specific data. However, its effectiveness is limited in data-scarce scenarios, often leading to suboptimal outcomes. Moreover, some methods were primarily developed for classification tasks (such as LwF (Li & Hoiem, 2017), IDBR (Huang et al., 2021)), which can not directly apply to text-to-text tasks.

2.2 CL Benchmarks in NLP

The most recognized CL benchmark for NLP encompasses five text classification datasets introduced by Zhang et al. (2015), including AG News, Amazon Reviews, Yelp Reviews, DBpedia, and Yahoo Answers. Building upon this, Razdaibiedina et al. (2023) proposed a long CL benchmark which fuses the aforementioned five datasets with an additional four from the GLUE benchmark (Wang et al., 2018), five from the SuperGLUE benchmark (Wang et al., 2019), and the IMDB dataset (Maas et al., 2011). While these datasets have been incorporated into the Flan collection (Chung et al., 2022) and are widely used for current SoTA LLMs, their ubiquity has rendered them less suitable as CL benchmarks for LLMs. However, this benchmark is limited in scope as it solely emphasizes Natural Language Generation (NLG) tasks and is restricted to English, thus lacking task diversity. In contrast, TRACE is a more varied and challenging benchmark, designed to test aligned LLMs across multiple sequential tasks. It assesses general capability, instruction adherence, and safety shifts. Recognizing that TRACE, like earlier benchmarks, may be incorporated into future pre-training, we acknowledge potential impacts on its long-term efficacy as a CL benchmark. Nonetheless, the current findings from TRACE provide essential insights into catastrophic forgetting in LLMs.

2.3 Chain-of-Thought

LLMs have shown advanced step-by-step reasoning capabilities, known as Chain-of-Thought (CoT) reasoning (Wei et al., 2022). Zero-shot-CoT (Kojima et al., 2022) illustrates the profound impact of simply prefacing a reasoning sequence with the sentence, ”Let’s think step by step.” Least-to-most (Zhou et al., 2022), actively prompts LLMs to segment complex problems into smaller tasks. ScienceQA (Lu et al., 2022) highlights the efficacy of CoT in LLMs, particularly beneficial for filling the void in datasets within the scientific realm. Fine-tune-CoT (Ho et al., 2022) exploits the prowess of extensive LLMs to generate reasoning samples, further fine-tuning smaller models. Despite these advancements, the application of CoT in continual learning remains unexplored. Our benchmark, TRACE, showcases that generating explanations can not only accelerate the learning process but also significantly mitigate the forgetting in their foundational capabilities.

3 Preliminaries

Continual learning (Ke & Liu, 2022; Wang et al., 2023a) focuses on developing learning algorithms to accumulate knowledge on non-stationary data. In supervised continual learning, a sequence of tasks $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$ arrive in a streaming fashion. Each task $\mathcal{D}_t = \{(\boldsymbol{x}_i^t, \boldsymbol{y}_i^t)\}_{i=1}^{n_t}$ contains a separate target dataset, where $\boldsymbol{x}_i^t \in \mathcal{X}_t$, $\boldsymbol{y}_i^t \in \mathcal{Y}_t$. A single model needs to adapt to them sequentially, with only access to $\mathcal{D}_t$ at the t-th task. In general, given a prediction model h_Θ parameterized by Θ, continual learning seeks to optimize for the following objective across all tasks:

$$\max_{\Theta} \sum_{k=1}^{T} \sum_{x,y \in \mathcal{D}_k} \log p_\Theta(y \mid x) \tag{1}$$

In this paper, we utilize overall performance (OP (Chaudhry et al., 2018)), forward transfer (FWT (Lopez-Paz & Ranzato, 2017)), and backward transfer (BWT (Lopez-Paz & Ranzato, 2017)) scores as the main metrics. After incrementally learning the t-th task, the model's score on the i-th task (where $i \leq t$) is denoted as $R^D_{t,i}$.

4 TRACE: A Comprehensive Benchmark for CL in LLMs

TRACE is designed to offer a comprehensive continual learning evaluation for LLMs. Illustrated in Figure 1, TRACE encompasses two primary components: a curated set of tasks tailored for continual learning, followed by an in-depth evaluation of an LLM's post-training capabilities. In this section, we detail TRACE's sequential tasks and introduce our evaluation metrics. In Section 4.4, we evaluate five models using TRACE and present our key findings.

4.1 Data Creation

There are three principles for the creation of TRACE. First, the datasets should be novel enough that most LLMs have not been trained on them. Second, they should be challenging for large language models. Third, a variety of tasks should be covered in our benchmark.

According to these three principles, in this section, we will provide a detailed introduction to the data collection process for each dataset. As these datasets have varying sizes, we create a balanced version by randomly sampling 5000 training examples and 2000 testing examples from the original datasets. As shown in Table 4, we get 40,000 training examples and 16,000 testing examples in total.

Domain-Specific. These tasks require specific knowledge, so models may perform poorly if they have not appeared frequently enough in the training data. We select three datasets, ScienceQA (Lu et al., 2022), FOMC (Shah et al., 2023) and MeetingBank (Hu et al., 2023).

Multi-lingual. The cross-lingual ability of large language models is limited due to vocabulary and pre-training corpus. For instance, LLaMA's vocabulary contains few Chinese tokens, affecting its efficiency with Chinese text. We select C-STANCE(Zhao et al., 2023) and 20Minuten(Rios et al., 2021) as multi-lingual datasets.

Code completion. Code completion is another challenging task to evaluate long context modeling ability(Bai et al., 2023), and it is one of the most widely used features in software development through IDEs. We select the line-level code completion task of CodeXGLUE(Lu et al., 2021) and Py150 (Lu et al., 2021) datasets.

Mathematical reasoning. Mathematical problems are always used to evaluate the reasoning ability of models. We include the first two tasks of NumGLUE(Mishra et al., 2022) because they are freshly collected data and intelligent modifications of already existing datasets. It is worth noting that both datasets have original labels consisting only of numbers, without associated inference processes.

4.2 CL Metrics Design

Unlike traditional continual learning benchmarks focused on sequential target tasks, evaluating aligned LLMs should also account for the preservation of their inherent capabilities. SoTA LLMs, through instruction tuning, exhibit impressive task-solving abilities. Aligning these models with human preferences further boosts their safety and usefulness. Hence, TRACE introduces a broader evaluation, including three unique metrics.

General Ability Delta is designed to assess the change in performance of an LLM on generic tasks after training on sequential target tasks. Let's consider a set of general tasks denoted as $\{G_1, ..., G_M\}$. The baseline performance of the initial LLM on the i-th general task is represented by $R^G_{0,i}$. After incrementally learning up to the t-th task, the score on the i-th general task becomes $R^G_{t,i}$. The "General Ability Delta" after training on the t-th task, represented as ΔR^G_t, is given by:

$$\Delta R^G_t = \frac{1}{M} \sum_{i=1}^{M} (R^G_{t,i} - R^G_{0,i}) \tag{2}$$

Instruction Following Delta measures the change in a model's ability to follow instructions after training on sequential tasks. Using a set of datasets, represented as $I_1, ..., I_N$, the initial LLM performance on the i-th task is $R^I_{0,i}$. After incremental learning to the t-th task, its score on the i-th task is $R^I_{t,i}$. The change, represented by ΔR^I_t, is computed as:

$$\Delta R^I_t = \frac{1}{N} \sum_{i=1}^{N} (R^I_{t,i} - R^I_{0,i}) \tag{3}$$

Safety Delta quantifies the change in a model's response safety after sequential training. Using a set of datasets designed for safety evaluation, denoted as $S_1, ..., S_L$, the initial safety metric on the i-th dataset is $R^S_{0,i}$. After training up to the t-th task, its score on the i-th dataset is $R^S_{t,i}$. The change after the t-th task, represented by ΔR^S_t, is computed as:

$$\Delta R^S_t = \frac{1}{L} \sum_{i=1}^{L} (R^S_{t,i} - R^S_{0,i}) \tag{4}$$

4.3 Experimental Setup

4.3.1 Models & Baselines

To evaluate the resilience of aligned LLMs from diverse training backgrounds and strategies, we select six backbones from three organizations: Meta: LLaMa-2-7B-Chat, LLaMa-2-13B-Chat, LLaMa-2-70B-Chat (Touvron et al., 2023), BaiChuan: Baichuan 2-7B-Chat (Baichuan, 2023), and Large Model Systems Organization: Vicuna-13B-V1.5, Vicuna-7B-V1.5 (Chiang et al., 2023).

We evaluate the performance of LLMs in a continual learning setting using four approaches—three requiring training and one not:

Sequential Full-Parameter Fine-Tuning (SeqFT): This method involves training all model parameters in sequence.

LoRA-based Sequential Fine-Tuning (LoraSeqFT): Only the low-rank LoRA matrices are fine-tuned, leaving the LLM backbone fixed (Hu et al., 2021). This method is chosen based on prior findings of reduced forgetting with ”Efficient Tuning” (Liu & Huang, 2023).

Replay-based Sequential Fine-Tuning (Replay): Replay, a common continual learning strategy, is employed for its simplicity and effectiveness. We incorporate alignment data from LIMA into the replay memory, replaying 10% of historical data.

In-Context Learning (ICL): Task demonstrations are supplied as part of the language prompt, acting as a form of prompt engineering (Brown et al., 2020). A 6-shot setting is used for our experiments.

4.3.2 Datasets

To evaluate a model's *general ability*, we assess across six key dimensions: **Factual Knowledge**: Using MMLU dataset (Hendrycks et al., 2020) with questions on 57 subjects, reporting 5-shot accuracy. **General Reasoning**: Evaluated with BBH (Suzgun et al., 2022), reporting EM scores with chain-of-thought prompts with 3-shot in-context examples. **Multilinguality**: Using TyDiQA (Clark et al., 2020), a multilingual QA benchmark across 11 languages, reporting 0-shot F1 scores. **Commonsense Reasoning**: Assessed with PIQA (Bisk et al., 2020), reporting 0-shot accuracy. **Code Generation**: Using MBPP (Austin et al., 2021), reporting 0-shot Pass@1. **Reading Comprehension**: Using BoolQ (Clark et al., 2019), reporting 0-shot accuracy.

For *instruction-following capability*, we use Self-instruct dataset (Wang et al., 2022a), comprising 175 user-oriented prompts, and the LIMA dataset (Zhou et al., 2023) , which assembles 300 prompts from community Q&A forums and manual examples.

For assessing changes in *safety*, we leverage the CoNa dataset (Bianchi et al., 2023). This corpus encompasses 178 expert-annotated samples, specifically curated to address instructions associated with hateful speech generation.

Table 1: Overall Performance (OP), Forward Transfer (FWT), and Backward Transfer (BWT) for all baseline models and 4 baseline methods.

	ICL	**SeqFT**			**LoRASeqFT**			**Replay**		
	OP	OP	BWT	FWT	OP	BWT	FWT	OP	BWT	FWT
LLaMA-2-7B-Chat	38.9	48.7	−8.3%	2.4%	12.7	−45.7%	0.8%	55.5	2.6%	−0.2%
LLaMA-2-13B-Chat	41.9	49.9	−7.0%	2.5%	28.0	−36.5%	1.5%	56.6	0.4%	2.2%
LLaMA-2-70B-Chat	48.4	-	-	-	45.2	−17.5%	1.7%	56.6	0.4%	1.9%
Vicuna-7B-V1.5	42.2	49.2	−8.4%	1.7%	33.4	−23.7%	0.9%	55.3	0.2%	0.5%
Vicuna-13B-V1.5	46.9	51.2	−5.9%	2.1%	31.6	−28.4%	1.1%	56.9	0.6%	2.4%
Baichuan-7B-Instruct	44.6	43.4	−15.4%	1.8%	43.8	−9.0%	1.0%	51.7	1.1%	−4.0%

4.3.3 Implementation Details

The detailed settings can be found in Appendix .1.

4.4 Main Results

4.4.1 Performance of Target Sequential Tasks

Table 1 showcases the performance of five distinct LLMs on TRACE benchmark, after their continual learning phase. From this evaluation, we can draw the following conclusions:

In-Context Learning (ICL): ICL methods generally perform lower than SeqFT and Replay methods. This suggests that the TRACE benchmark is indeed challenging, and LLMs can't readily identify solutions just through simple demonstrations.

Replay Performance: Among all the baselines, Replay achieved the highest OP score. With its BWT score being positive, it indicates that Replay effectively retains its performance on sequential tasks without significant forgetting. This makes Replay a straightforward and efficient strategy in a continual learning context.

Full Parameter Training vs. LoRA: Full parameter training demonstrates better task-specific adaptability compared to LoRA, with a smaller BWT score. For instance, LLaMA-2-7B-Chat's SeqFT OP(BWT) is 48.7 (-8.3%), while LoRASeqFT stands at 12.7 (-45.7%).

Catastrophic Forgetting in LLaMA-2-70B-Chat: Despite achieving a high overall performance (OP) score of 45.2, the llama2-70B-Chat model exhibits significant catastrophic forgetting, as evidenced by its BWT of -17.5%. This substantial decline in retaining previously learned information indicates a vulnerability of larger models to catastrophic forgetting in continual learning scenarios.

4.4.2 Variation of General Ability

Table 2 presents the evaluations of various LLM models concerning general abilities. The degree of general ability forgetting in LLMs can be analyzed from three perspectives. For a more detailed evaluation, refer to the Appendix.

From the Model Perspective: **1)** Nearly all models display a negative General Ability Delta, indicating a general decline in overall capabilities after continual learning. **2)** Both larger and smaller models experience significant forgetting in general abilities. For instance, the General Ability Delta for llama2-7B-chat-LoraSeq stands at -7.88, whereas llama2-70B-chat-LoraSeq is -10.04.

From the Task Perspective: **1)** Despite the presence of CoT prompts, there is a noticeable decline in math and reasoning abilities across all models, suggesting that these abilities are highly sensitive to new task learning. **2)** Excluding the llama2-7b model, most models exhibit a significant drop in performance on MMLU, suggesting a gradual loss of factual knowledge through continual learning. **3)** TydiQA task sees a general boost post-training, possibly due to the inclusion of Chinese and German datasets in our sequential tasks. Even more intriguing is the observed enhancement (and some declines) in other languages on TydiQA, suggesting potential cross-linguistic transfer characteristics. **4)** Performance shifts on PIQA for most models are subtle, indicating the relative robustness of commonsense knowledge during continual learning.

Table 2: Comparison of the general language understanding and reasoning abilities. blue means increase, while red means decrease.

	MMLU (factuality)	GSM (math)	BBH (reasoning)	TydiQA (multilinguality)	BoolQ (comprehension)	PIQA (commonsense)	MBPP (code)	ΔR_t^G
	ACC (5-shot)	EM (8-shot, CoT)	EM (3-shot, CoT)	F1 (1-shot, GP)	ACC (0-shot)	ACC (0-shot)	Pass@1 (0-shot)	
LLaMA-2-7B-Chat	46.56	26.08	40.23	23.47	70.55	76.22	20.4	0
LLaMA-2-7B-Chat-Seq	46.43	3.49	30.11	33.23	77.89	76.5	0	−5.12
LLaMA-2-7B-Chat-LoraSeq	42.28	14.71	33.61	21.72	53.43	75.19	7.4	−7.88
LLaMA-2-7B-Chat-Replay	47.04	3.03	36.61	31.57	75.75	75.3	4.4	−4.26
LLaMA-2-13B-Chat	54.61	43.14	49.70	27.65	81.5	78.24	27.2	0
LLaMA-2-13B-Chat-Seq	41.88	2.12	19.47	32.27	82.08	77.15	1.2	−15.13
LLaMA-2-13B-Chat-LoraSeq	50.63	24.72	38.98	26.93	68.96	78.02	10.4	−9.06
LLaMA-2-13B-Chat-Replay	47.72	2.96	36.52	32.52	82.45	76.88	7.2	−10.83
LLaMA-2-70B-Chat	63.80	59.35	60.81	46.04	88.30	80.60	35.64	0
LLaMA-2-70B-Chat-LoraSeq	60.56	44.79	49.77	43.25	72.15	77.47	16.30	−10.04
LLaMA-2-70B-Chat-Replay	62.48	47.34	52.34	44.12	88.68	77.69	12.57	−7.05
Baichuan2-7B-Instruct	53.80	33.21	35.66	20.64	77.09	74.05	22.4	0
Baichuan2-7B-Instruct-Seq	46.92	4.25	37.45	35.20	79.08	74.21	1.2	−5.5
Baichuan2-7B-Instruct-LoraSeq	52.14	22.74	27.53	30.99	75.23	74.86	8.6	−3.53
Baichuan2-7B-Instruct-Replay	45.72	8.19	35.61	34.65	80.06	72.69	6.4	−4.79
Vicuna-7B-V1.5	51.28	23.65	43.32	22.38	78.56	77.42	12.6	0
Vicuna-7B-V1.5-Seq	49.46	3.87	39.25	33.92	77.74	75.73	1.2	−4.0
Vicuna-7B-V1.5-LoraSeq	48.37	18.89	28.16	25.84	67.24	76.23	0	−7.44
Vicuna-7B-V1.5-Replay	47.20	4.78	39.26	31.86	78.92	80.13	2.0	−4.76
Vicuna-13B-V1.5	56.16	36.09	51.29	24.89	82.45	78.89	2.6	0
Vicuna-13B-V1.5-Seq	37.93	2.81	35.23	36.86	83.43	77.86	0	−8.32
Vicuna-13B-V1.5-LoraSeq	52.46	22.14	41.22	27.86	67.71	77.53	0.4	−6.15
Vicuna-13B-V1.5-Replay	48.73	3.11	42.94	39.60	84.71	77.53	0.6	−0.52

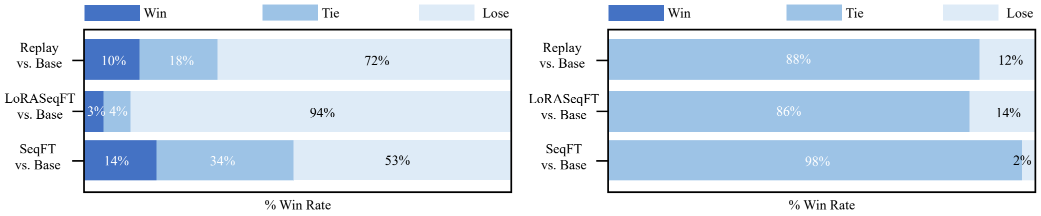
From the Method Perspective: **1)** The Replay method proves beneficial in preserving reasoning and factuality skills. Especially for larger models, the mitigation of forgetting through Replay is more pronounced. For instance, for LLaMA-2-7B-Chat, Replay offers a 6.5 EM score boost compared to methods without Replay, while for LLaMA-2-13B-Chat, the increase is 17.1 EM score.

4.4.3 Instruction Following Ability Analysis

We evaluate the instruction-following ability of models based on two foundation models: LLaMA-2-7B-Chat and LLaMA-2-13B-Chat. Figure 2 (a) illustrates the win rate % for instruction following sequentially trained LLMs and their original versions. Here, the win rate can be approximated as an indicator for the Instruction-following delta. It's evident that all three training methods exhibit a marked decline in instruction-following capabilities compared to their initial versions, with the decline being most pronounced in the LoRA method. Therefore, be cautious when exploring approaches like LoRA for continual learning in LLMs.

4.4.4 Safety Analysis

We test the safety of answers from models LLaMA-2-7B-Chat and LLaMA-2-13B-Chat. Figure 2 (b) shows the win rate % for instruction following between the new LLMs and their starting versions. Here, the win rate can be used as a measure for the Safety Delta. Compared to the original models, most answers were rated as 'Tie'. This suggests that the safety of the model's answers is largely unaffected by continual learning on general tasks.



(a) Helpful evaluation

(b) Safety evaluation

Figure 2: GPT-4 evaluation with llama-13b-chat, comparing 3 different baselines (Replay, LoRA and Sequential) to the base model across tasks including helpful and safety.

4.5 Influencing Factors of Forgetting

4.5.1 Data Quantity & Training Steps

Figure 3 shows the performance on target tasks for continual learning datasets with different data volumes and training steps. For LLaMA-2-7B-Chat's SeqFT, we tested with 500, 1000, and 5000 samples from each dataset, training them for 1, 3, 5, 10 epochs. Performance improves as data volume grows, indicating at least 5000 samples from the TRACE-selected datasets are needed for full fitting. Additionally, performance improves with up to 5 training epochs, confirming our baseline epoch setting balances target task optimization and retaining existing capabilities.

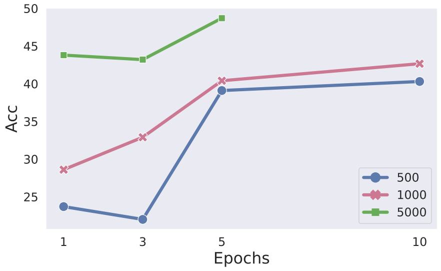


Figure 3: Performance evaluation of LLaMA-2-7B-Chat's SeqFT on the TRACE benchmark across varying sample sizes (500, 1000, 5000) and training epochs (1, 3, 5, 10 (except for 5000)).

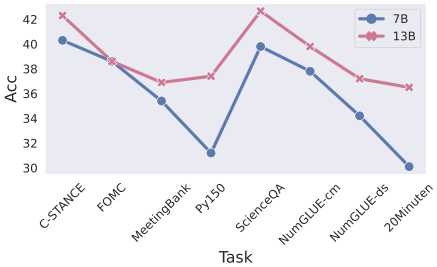


Figure 4: Evolution of LLMs' reasoning capabilities post-training on different tasks, measured using the BBH performance metric. We report the results of LLaMA-2-7B-chat and LLaMA-2-13B-chat.

4.5.2 Decoupling the Impact of Different Tasks

From the results in section 4.4.2, it's evident that post-training LLMs on our benchmark, their innate reasoning and mathematical abilities see a significant dip. This brings forth the question: *How exactly does the reasoning capability of LLMs transform during the continual learning process?*

Figure 4 tracks the reasoning ability (assessed via BBH performance) after the completion of each training task. Intriguingly, we observe a surge in the model's reasoning prowess post-training on the ScienceQA task, while it declines for other tasks. Notably, even though the two tasks from NumGLUE are mathematically inclined, their answers do not provide a clear reasoning path. In contrast, ScienceQA does offer such a pathway in its answers. This observation suggests the potential advantage of incorporating reasoning paths during training to preserve and perhaps even enhance the model's reasoning capability.

5 Reasoning-augmented Continual learning

Drawing from our earlier findings, which underscore the unique capabilities of LLMs, we are inspired to reconsider how we approach their training. Instead of treating LLMs as traditional models and inundating them with large volumes of data to fit a task's distribution, we leverage their inherent abilities for rapid task transfer. With these insights as our foundation, we formulate the Reasoning-augmented Continual Learning (RCL) approach. RCL forms reasoning paths on new datasets, aiming to not only preserve LLMs' reasoning capabilities but also to enhance their task transfer and output clarity.

As depicted in Figure 5, RCL involves two phases: automated reasoning annotation and sequential training on the augmented datasets. Domain experts create prompts for each task. Three samples per task are manually annotated. GPT-4, using these prompts, generates reasoning paths for every entry. Reasoning is verified against ground truth. We rely on machine-generated answers due to cost concerns and the consistency of the LM-generated text. To validate reasoning quality, we manually inspect outputs, achieving a 94% approval rate on a 100-sample check, highlighting GPT-4's reliability. Following this, supervised training is conducted on the target LLM, keeping hyperparameter settings consistent with baselines.

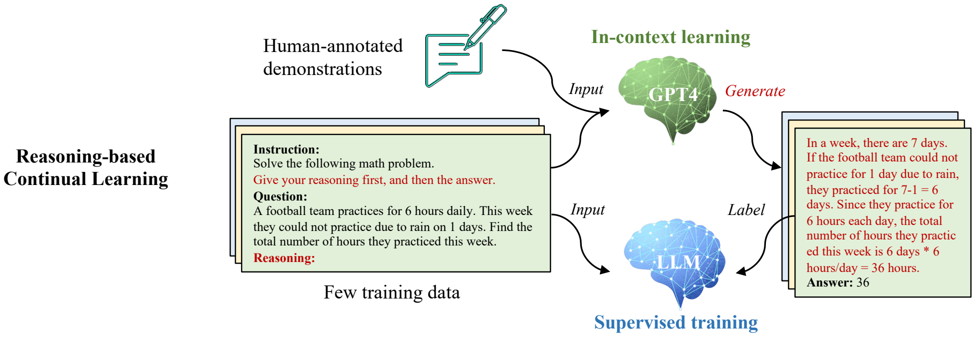

Human-annotated
demonstrations
In-context learning
Input
GPT4
Generate
Reasoning-based
Continual Learning
Instruction:
Solve the following math problem.
Give your reasoning first, and then the answer.
Question:
A football team practices for 6 hours daily. This week they could not practice due to rain on 1 days. Find the total number of hours they practiced this week.
Reasoning:
Few training data
Input
Label
LLM
In a week, there are 7 days. If the football team could not practice for 1 day due to rain, they practiced for 7-1 = 6 days. Since they practice for 6 hours each day, the total number of hours they practic ed this week is 6 days * 6 hours/day = 36 hours.
Answer: 36
Supervised training

Figure 5: An overview of Reasoning-augmented continual learning method. Our method unfolds in two stages: 1) Automatic annotation of sample reasoning paths using GPT-4. 2) Continual learning on reasoning-augmented datasets.

5.1 Performance on Sequential Tasks

Table 3 provides comparisons of the performance of our RCL method against other techniques with LLaMA-2-7B-Chat. **O-Lora** (Wang et al., 2023b) and **PP** Razdaibiedina et al. (2023) are two continual learning methods. Through an ablation study contrasting single-task training (SingleFT) with multi-task training, and assessing the impact of reasoning-augmented data, we observed that integrating reasoning pathways into the data consistently boosts performance over the original dataset.

Moreover, our approach, even when trained with just 500 samples, achieves comparable results to the SeqFT method trained on 5000 samples. Furthermore, by leveraging fewer datasets and training steps in our method, we would mitigate the decline in LLMs' inherent capabilities.

Table 3: Comparison of RCL with different baselines. Except for methods specifically noted with sample sizes, all methods are trained with 0.5k samples. **Single FT** refers to fine-tuning the model on a single task, **Re** refers to reasoning-augmented, and **MT** refers to Multi-task training.

Methods	**OP**	**WT**	**FWT**
ICL	39.5	-	-
ICL+Re	41.1	-	-
O-Lora	41.3	−6.2%	−6.3%
PP	46.2	−2.3%	3.1%
SeqFT(0.5k)	23.0	−19%	0.3%
SeqFT(5k)	48.7	−8.3%	2.4%
RCL(0.5k)	46.6	−13%	2.7%
RCL(1k)	46.2	−13.1%	2.5%
RCL(2k)	50.0	−8.4%	3.4%
RCL(5k)	**51.9**	−6.5%	3.2%
SingleFT	57.6	-	-
SingleFT+Re	58.1	-	-
MT w/o. Re	52.3	-	-
MT w. Re	58.2	-	-

5.2 Impacts on General Ability

We report the performance of general abilities in Figure 6. We can conclude that RCL reaches comparable performance with SeqFT and Replay method on MMLU, TydiQA, BoolQA and PIQA though only. However, RCL stands out in reasoning tasks such as GSM and BBH. For instance, for the GSM task, RCL outperforms SeqFT and Replay by 12.7 and 13.2 points respectively, showing the advantages of providing reasoning paths in maintaining the reasoning abilities of models. Besides, combining RCL with replay further improves its performance on reasoning tasks.

5.3 Impacts on Instruction-Following and Safety

The impact of incorporating RCL on instruction-following capabilities is presented in Table 5. It's evident that RCL enhances the model's ability to follow instructions by 8% and 5% compared to SeqFT and Replay, respectively.

6 Conclusion

Existing continual learning benchmarks are insufficient in thoroughly evaluating LLMs, due to their oversimplification and lack of metrics for critical capabilities like instruction following and safety. To tackle this, we introduce TRACE, a comprehensive benchmark with eight challenging tasks and well-rounded metrics. Our experiments show that for LLMs, catastrophic forgetting still remains, and a clear drop in general abilities is observed during continual learning. Besides, our Reasoning-augmented Continual Learning (RCL) method highlights the importance of using reasoning in training while alleviating above phenomena. We believe this area is very important and hope our work lays a solid foundation for future studies.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *ArXiv*, abs/2108.07732, 2021. URL https://api.semanticscholar.org/CorpusID:237142385.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding, 2023.

Baichuan. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023. URL https://arxiv.org/abs/2309.10305.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*, 2023.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

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, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff 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. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. URL https://api.semanticscholar.org/CorpusID:218971783.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 532–547, 2018.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416, 2022. URL https://api.semanticscholar.org/CorpusID:253018554.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8: 454–470, 2020.

OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass, 2023.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773. PMLR, 2020.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*, 2022.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL `https://api.semanticscholar.org/CorpusID:235458009`.

Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. Meetingbank: A benchmark dataset for meeting summarization, 2023.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. Continual learning for text classification with information disentanglement based regularization, 2021.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*, 2023.

Zixuan Ke and Bin Liu. Continual learning of natural language processing tasks: A survey. *ArXiv*, abs/2211.12701, 2022.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*, 2023.

Zhizhong Li and Derek Hoiem. Learning without forgetting, 2017.

Lei Liu and J. Huang. Prompt learning to mitigate catastrophic forgetting in cross-lingual transfer for open-domain dialogue generation. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023. URL `https://api.semanticscholar.org/CorpusID:258676338`.

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

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. Codexglue: A machine learning benchmark dataset for code understanding and generation, 2021.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, A. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics*, 2011. URL https://api.semanticscholar.org/CorpusID:1428702.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks, 2022.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

Chengwei Qin and Shafiq Joty. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5, 2022.

Yujia Qin, Shi Liang, Yining Ye, Kunlun Zhu, Lan Yan, Ya-Ting Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Marc H. Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. *ArXiv*, abs/2307.16789, 2023. URL https://api.semanticscholar.org/CorpusID:260334759.

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

Annette Rios, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. A new dataset and efficient baselines for document-level text simplification in German. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pp. 152–161, Online and in Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.newsum-1.16. URL https://aclanthology.org/2021.newsum-1.16.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned language models are continual learners. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://api.semanticscholar.org/CorpusID:252815378.

Agam Shah, Suvan Paturi, and Sudheer Chava. Trillion dollar words: A new financial dataset, task & market analysis, 2023.

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

Nigar M Shafiq Surameery and Mohammed Y Shakor. Use chat gpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, 3(01):17–22, 2023.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

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

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *ArXiv*, abs/2302.00487, 2023a.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning, 2023b.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022a.

Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, and Wenqiu Zeng. Rehearsal-free continual language learning via efficient parameter isolation. *ArXiv*, 2023c.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning, 2022b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015. URL `https://api.semanticscholar.org/CorpusID:368182`.

Chenye Zhao, Yingjie Li, and Cornelia Caragea. C-STANCE: A large dataset for Chinese zero-shot stance detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13369–13385, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.747. URL `https://aclanthology.org/2023.acl-long.747`.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

APPENDIX

.1 IMPLEMENTATION DETAILS

During the training phase, for the baselines without LoRA adapters, we consistently trained with 5000 samples with a constant learning rate of 1e-5. For the datasets we used (C-STANCE, FOMC, MeetingBank, Py150, ScienceQA, NumGLUE-cm, NumGLUE-ds, 20Minuten), we train for 1, 1, 5, 5, 1, 5, 5, 5 epochs respectively. While for the baselines with LoRA adapters, we trained with

5000 samples with a constant learning rate of 1e-4 for 5, 3, 7, 5, 3, 5, 5, 7 epochs respectively. Our training settings incorporated a weight decay set to 0, and a batch size of 128. For the testing phase, we use a temperature of 0.1. All our training and inference experiments were conducted on a machine equipped with 8x80G Nvidia A100, and were implemented using DeepSpeed repository. All models are trained on 8 A100 GPUs with 80G memory with full parameters fine-tuning. We leave the exploration of large models like LLaMa-2-65B-Chat for future work due to the current hardware limits.

All general benchmark evaluations were conducted using the Open-Compass toolkit (Contributors, 2023), adopting its default configuration.

.2 TRACE DATASET STATISTICS

In this section, we represent the overview of dataset statistics, including source, average lenght, metric, language and number of samples of each dataset in TRACE benchmark.

Table 4: An overview of dataset statistics in TRACE. 'Source' indicates the context's origin. 'Avg len' represents word count for English, German, and code datasets, and character count for Chinese. 'SARI' is a score specific to simplification.

Dataset	Source	Avg len	Metric	Language	#data
Domain-specific					
ScienceQA	Science	210	Accuracy	English	5,000
FOMC	Finance	51	Accuracy	English	5,000
MeetingBank	Meeting	2853	ROUGE-L	English	5,000
Multi-lingual					
C-STANCE	Social media	127	Accuracy	Chinese	5,000
20Minuten	News	382	SARI	Germany	5,000
Code completion					
Py150	Github	422	Edim similarity	Python	5,000
Mathematical reasoning					
NumGLUE-cm	Math	32	Accuracy	English	5,000
NumGLUE-ds	Math	21	Accuracy	English	5,000

.3 SUPPLEMENTARY TABLES & FIGURES

Table 5: Instruction-following abilities of SeqFT, Replay and RCL. The model we use is LLaMA-2-7b-Chat.

	Win	Tie	Loss
SeqFT	12%	30%	58%
Replay	15%	31%	55%
RCL	20%	30%	50%

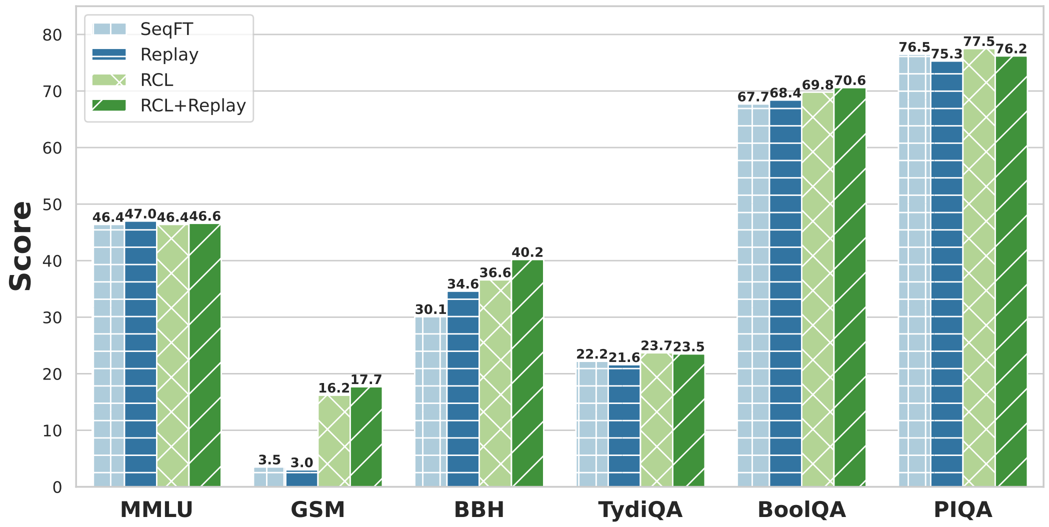


Figure 6: OpenCompass Evaluation Results. **RCL+Replay** refers to combining our RCL method with replay method. The model we use is LLaMA-2-7b-Chat.

.4 Detailed Experiments Results

In this section, we report the detailed experiment results in our paper. The model includes Baichuan-7b,LLaMA2-7b-chat, LLaMA2-13b-chat, Vicuna-7b and Vicuna-13b. The results are shown in Table 6 30.

.4.1 In-Context Learning

Table 6 represents the performance of different models with in-context learning.

Table 6: Detailed results of in-context learning of different large language models.

Task/Model	Baichuan-7b	LLaMA2-7b-chat	LLaMA2-13b-chat	LLaMA2-70b-chat	Vicuna-7b	Vicuna-13b
C-STANCE	0.58	0.4	0.366	0.45	0.403	0.57
FOMC	0.63	0.483	0.519	0.567	0.551	0.61
MeetingBank	0.225	0.198	0.221	0.345	0.223	0.229
Py150	0.586	0.522	0.539	0.577	0.529	0.585
ScienceQA	0.68	0.628	0.689	0.724	0.695	0.7
NumGLUE-cm	0.271	0.284	0.407	0.436	0.284	0.347
NumGLUE-ds	0.23	0.203	0.218	0.376	0.302	0.33
20Minuten	0.366	0.395	0.395	0.395	0.392	0.378
average	0.446	0.389	0.419	0.484	0.422	0.469

.4.2 SeqFT method

Table 7 - 12 shows the detailed performance of different models of each round during the continual learning. SeqFT represents sequential fine-tuning.

Table 7: Detailed results of sequential fine-tuning of Baichuan-7b.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.62	0.629	0.663	0.621	0.531	0.55	0.588	0.579
FOMC	-	0.681	0.353	0.318	0.02	0.363	0.347	0.335
MeetingBank	-	-	0.442	0.351	0.371	0.379	0.389	0.364
Py150	-	-	-	0.626	0.562	0.586	0.589	0.58
ScienceQA	-	-	-	-	0.77	0.68	0.5	0.44
NumGLUE-cm	-	-	-	-	-	0.358	0.247	0.284
NumGLUE-ds	-	-	-	-	-	-	0.64	0.475
20Minuten	-	-	-	-	-	-	-	0.415
average								0.434
BWT								-0.154
FWT								0.018

Table 8: Detailed results of sequential fine-tuning of LLaMA-7b-chat.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.5	0.456	0.448	0.453	0.435	0.442	0.436	0.454
FOMC	0.54	0.735	0.67	0.658	0	0.595	0.577	0.609
MeetingBank	0.172	0.098	0.523	0.459	0.433	0.446	0.442	0.457
Py150	0.218	0.282	0.185	0.58	0.459	0.509	0.508	0.512
ScienceQA	0.12	0.13	0.15	0.17	0.764	0.636	0.45	0.637
NumGLUE-cm	0.173	0.012	0.074	0.062	0	0.383	0.247	0.272
NumGLUE-ds	0.1	0	0.04	0.11	0	0.03	0.582	0.548
20Minuten	0.022	0.021	0.018	0.024	0.017	0.019	0.017	0.408
average								0.487
BWT								-0.083
FWT								0.024

Table 9: Detailed results of sequential fine-tuning of LLaMA-13b-chat.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.469	0.465	0.47	0.477	0.463	0.466	0.45	0.5
FOMC	0.587	0.754	0.738	0.748	0.03	0.721	0.71	0.717
MeetingBank	0.232	0.165	0.533	0.51	0.375	0.421	0.385	0.351
Py150	0.246	0.265	0.213	0.568	0.538	0.537	0.541	0.547
ScienceQA	0.223	0.256	0.197	0.244	0.8	0.655	0.241	0.55
NumGLUE-cm	0.198	0.121	0.089	0	0	0.333	0.284	0.296
NumGLUE-ds	0.156	0.034	0	0.045	0	0.245	0.618	0.622
20Minuten	0.032	0.034	0.026	0.024	0.025	0.034	0.018	0.408
average								0.499
BWT								-0.07
FWT								0.025

Table 10: Detailed results of sequential fine-tuning of LLaMA-7b.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.56	0.45	0.45	0.42	0.42	0.44	0.48	0.49
FOMC	-	0.75	0.53	0.53	0	0.72	0.72	0.38
MeetingBank	-	-	0.427	0.233	0.255	0.231	0.245	0.226
Py150	-	-	-	0.617	0.578	0.565	0.589	0.579
ScienceQA	-	-	-	-	0.1	0	0	0
NumGLUE-cm	-	-	-	-	-	0.32	0.197	0.197
NumGLUE-ds	-	-	-	-	-	-	0.66	0.37
20Minuten	-	-	-	-	-	-	-	0.188
average								0.303
BWT								-0.17
FWT								0.005

Table 11: Detailed results of sequential fine-tuning of Vicuna-7b.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.532	0.451	0.439	0.448	0.149	0.477	0.47	0.476
FOMC	-	0.738	0.732	0.744	0	0.605	0.567	0.675
MeetingBank	-	-	0.519	0.447	0.443	0.427	0.417	0.439
Py150	-	-	-	0.577	0.384	0.486	0.482	0.482
ScienceQA	-	-	-	-	0.773	0.7	0.608	0.649
NumGLUE-cm	-	-	-	-	-	0.407	0.247	0.296
NumGLUE-ds	-	-	-	-	-	-	0.578	0.517
20Minuten	-	-	-	-	-	-	-	0.403
average								0.492
BWT								-0.084
FWT								0.017

Table 12: Detailed results of sequential fine-tuning of Vicuna-13b.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.527	0.43	0.471	0.497	0.374	0.468	0.469	0.484
FOMC	-	0.741	0.739	0.731	0	0.754	0.678	0.714
MeetingBank	-	-	0.549	0.532	0.53	0.491	0.427	0.412
Py150	-	-	-	0.564	0.54	0.546	0.538	0.552
ScienceQA	-	-	-	-	0.79	0.616	0.586	0.633
NumGLUE-cm	-	-	-	-	-	0.346	0.309	0.358
NumGLUE-ds	-	-	-	-	-	-	0.622	0.572
20Minuten	-	-	-	-	-	-	-	0.41
average								0.517
BWT								-0.059
FWT								0.021

.4.3 SeqLoraFT method

Table 12 16 shows the detailed performance of different models of each round during the continual learning. SeqLoRAFT represents sequential fine-tuning with LoRA adapters.

Table 13: Detailed results of sequential fine-tuning of Baichuan-7b with LoRA adapters.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.613	0.601	0.597	0.584	0.506	0.504	0.53	0.477
FOMC	-	0.652	0.604	0.591	0.602	0.588	0.587	0.417
MeetingBank	-	-	0.345	0.334	0.333	0.343	0.34	0.337
Py150	-	-	-	0.588	0.472	0.539	0.517	0.472
ScienceQA	-	-	-	-	0.641	0.68	0.625	0.63
NumGLUE-cm	-	-	-	-	-	0.457	0.432	0.407
NumGLUE-ds	-	-	-	-	-	-	0.43	0.36
20Minuten	-	-	-	-	-	-	-	0.407
average								0.438
BWT								-0.090
FWT								0.01

Table 14: Detailed results of sequential fine-tuning of LLaMA-7b-chat with LoRA adapters.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.511	0.45	0.412	0.373	0.133	0.391	0.294	0.277
FOMC	-	0.713	0.55	0.452	0	0.421	0.341	0.24
MeetingBank	-	-	0.51	0.212	0.151	0.067	0.037	0.121
Py150	-	-	-	0.578	0.004	0.495	0.452	0.004
ScienceQA	-	-	-	-	0.68	0.645	0.535	0
NumGLUE-cm	-	-	-	-	-	0.37	0.235	0
NumGLUE-ds	-	-	-	-	-	-	0.486	0
20Minuten	-	-	-	-	-	-	-	0.37
average								0.127
BWT								-0.457
FWT								0.008

Table 15: Detailed results of sequential fine-tuning of LLaMA-13b with LoRA adapters.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.62	0.36	0.432	0.491	0.18	0.42	0.411	0.124
FOMC	-	0.743	0.681	0.63	0.53	0.605	0.579	0
MeetingBank	-	-	0.484	0.264	0.201	0.147	0.032	0.122
Py150	-	-	-	0.581	0.397	0.488	0.497	0.249
ScienceQA	-	-	-	-	0.75	0.729	0.714	0.68
NumGLUE-cm	-	-	-	-	-	0.58	0.296	0.259
NumGLUE-ds	-	-	-	-	-	-	0.62	0.386
20Minuten	-	-	-	-	-	-	-	0.417
average								0.28
BWT								-0.365
FWT								0.015

Table 16: Detailed results of sequential fine-tuning of LLaMA-70b with LoRA adapters.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.53	0.498	0.422	0.49	0.04	0.44	0.457	0.228
FOMC	-	0.641	0.532	0.66	0.01	0.64	0.67	0.19
MeetingBank	-	-	0.528	0.22	0.21	0.146	0.264	0.181
Py150	-	-	-	0.633	0.346	0.651	0.671	0.595
ScienceQA	-	-	-	-	0.81	0.79	0.89	0.805
NumGLUE-cm	-	-	-	-	-	0.543	0.531	0.518
NumGLUE-ds	-	-	-	-	-	-	0.74	0.68
20Minuten	-	-	-	-	-	-	-	0.42
average								0.452
BWT								-0.175
FWT								0.015

Table 17: Detailed results of sequential fine-tuning of Vicuna-7b with LoRA adapters.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.514	0.452	0.433	0.446	0	0.344	0.089	0.141
FOMC	-	0.715	0.48	0.427	0	0.272	0.304	0.29
MeetingBank	-	-	0.5	0.113	0.144	0.026	0.011	0.07
Py150	-	-	-	0.573	0.222	0.47	0.452	0.413
ScienceQA	-	-	-	-	0.67	0.632	0.53	0.6
NumGLUE-cm	-	-	-	-	-	0.407	0.37	0.259
NumGLUE-ds	-	-	-	-	-	-	0.545	0.492
20Minuten	-	-	-	-	-	-	-	0.409
average								0.334
BWT								-0.237
FWT								0.009

Table 18: Detailed results of sequential fine-tuning of Vicuna-13b with LoRA adapters.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.524	0.504	0.394	0.385	0.389	0.347	0.329	0.07
FOMC	-	0.74	0.68	0.616	0.188	0.62	0.438	0.04
MeetingBank	-	-	0.495	0.24	0.157	0.132	0.08	0.14
Py150	-	-	-	0.6	0.368	0.52	0.491	0.256
ScienceQA	-	-	-	-	0.77	0.75	0.732	0.74
NumGLUE-cm	-	-	-	-	-	0.407	0.346	0.346
NumGLUE-ds	-	-	-	-	-	-	0.569	0.52
20Minuten	-	-	-	-	-	-	-	0.413
average								0.316
BWT								-0.284
FWT								0.011

.4.4 Replay method

Table 17 21 shows the detailed performance of different models of each round during the continual learning with replay data.

Table 19: Detailed results of continual learning of Baichuan-7b with replay data.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.57	0.55	0.56	0.63	0.6	0.64	0.62	0.61
FOMC	-	0.69	0.64	0.64	0.65	0.65	0.66	0.61
MeetingBank	-	-	0.445	0.457	0.449	0.466	0.461	0.482
Py150	-	-	-	0.546	0.577	0.577	0.613	0.583
ScienceQA	-	-	-	-	0.58	0.51	0.54	0.57
NumGLUE-cm	-	-	-	-	-	0.321	0.346	0.333
NumGLUE-ds	-	-	-	-	-	-	0.5	0.55
20Minuten	-	-	-	-	-	-	-	0.405
average								0.517
BWT								0.011
FWT								-0.04

Table 20: Detailed results of continual learning of LLaMA-7b-chat with replay data.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.471	0.487	0.485	0.5	0.486	0.475	0.493	0.5
FOMC	-	0.734	0.769	0.785	0.807	0.781	0.785	0.8
MeetingBank	-	-	0.499	0.496	0.507	0.494	0.492	0.51
Py150	-	-	-	0.543	0.561	0.546	0.552	0.55
ScienceQA	-	-	-	-	0.763	0.78	0.78	0.785
NumGLUE-cm	-	-	-	-	-	0.358	0.309	0.37
NumGLUE-ds	-	-	-	-	-	-	0.486	0.52
20Minuten	-	-	-	-	-	-	-	0.406
average								0.555
BWT								0.026
FWT								-0.002

Table 21: Detailed results of continual learning of LLaMA-13b-chat with replay data.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.5	0.496	0.497	0.493	0.52	0.503	0.5	0.51
FOMC	-	0.778	0.803	0.805	0.792	0.789	0.785	0.813
MeetingBank	-	-	0.484	0.495	0.515	0.499	0.503	0.482
Py150	-	-	-	0.523	0.549	0.532	0.534	0.523
ScienceQA	-	-	-	-	0.816	0.8	0.804	0.792
NumGLUE-cm	-	-	-	-	-	0.358	0.407	0.396
NumGLUE-ds	-	-	-	-	-	-	0.628	0.606
20Minuten	-	-	-	-	-	-	-	0.407
average								0.566
BWT								0.004
FWT								0.022

Table 22: Detailed results of continual learning of LLaMA-70b-chat with replay data. Considering computational constraints, we also conduct this experiment using LoRA.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.519	0.506	0.61	0.62	0.63	0.58	0.69	0.67
FOMC	-	0.669	0.803	0.805	0.792	0.789	0.785	0.813
MeetingBank	-	-	0.484	0.495	0.515	0.499	0.503	0.482
Py150	-	-	-	0.523	0.549	0.532	0.534	0.523
ScienceQA	-	-	-	-	0.816	0.8	0.804	0.792
NumGLUE-cm	-	-	-	-	-	0.358	0.407	0.396
NumGLUE-ds	-	-	-	-	-	-	0.628	0.606
20Minuten	-	-	-	-	-	-	-	0.407
average								0.566
BWT								-0.04
FWT								0.022

Table 23: Detailed results of continual learning of Vicuna-7b with replay data.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.5	0.528	0.512	0.519	0.518	0.519	0.515	0.524
FOMC	-	0.747	0.803	0.794	0.805	0.795	0.801	0.806
MeetingBank	-	-	0.512	0.483	0.516	0.516	0.492	0.496
Py150	-	-	-	0.525	0.569	0.553	0.551	0.551
ScienceQA	-	-	-	-	0.77	0.776	0.772	0.767
NumGLUE-cm	-	-	-	-	-	0.396	0.322	0.309
NumGLUE-ds	-	-	-	-	-	-	0.554	0.563
20Minuten	-	-	-	-	-	-	-	0.405
average								0.553
BWT								0.002
FWT								0.005

Table 24: Detailed results of continual learning of Vicuna-13b with replay data.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.56	0.58	0.616	0.62	0.616	0.637	0.629	0.629
FOMC	-	0.736	0.76	0.76	0.788	0.771	0.76	0.76
MeetingBank	-	-	0.464	0.505	0.468	0.441	0.473	0.451
Py150	-	-	-	0.544	0.559	0.563	0.591	0.554
ScienceQA	-	-	-	-	0.71	0.699	0.674	0.71
NumGLUE-cm	-	-	-	-	-	0.42	0.358	0.358
NumGLUE-ds	-	-	-	-	-	-	0.667	0.68
20Minuten	-	-	-	-	-	-	-	0.41
average								0.569
BWT								0.006
FWT								0.024

.4.5 RCL METHOD

Table 22 shows the detailed performance of LLaMA2-7b-chat of each round during the continual learning. RCL represents reasoning-based continual learning.

Table 25: Detailed results of RCL of LLaMA2-7b-chat.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.614	0.428	0.464	0.486	0.5	0.472	0.452	0.522
FOMC	-	0.621	0.476	0.002	0.563	0.542	0.534	0.516
MeetingBank	-	-	0.497	0.431	0.329	0.363	0.332	0.343
Py150	-	-	-	0.563	0.513	0.521	0.528	0.527
ScienceQA	-	-	-	-	0.72	0.624	0.6	0.598
NumGLUE-cm	-	-	-	-	-	0.691	0.494	0.469
NumGLUE-ds	-	-	-	-	-	-	0.566	0.354
20Minuten	-	-	-	-	-	-	-	0.402
average								0.466
BWT								-0.135
FWT								0.036

.4.6 Different amounts of data and training steps

Table 23-25 shows the performance of LLaMA2-7b-chat with different number of data and training epochs.

Table 26: Performance of LLaMA-7b-chat after training on all of the sequential tasks for different epochs. Each dataset is sampled with 500 examples.

Task/Number of epochs	1	3	5
C-STANCE	0.24	0.38	0.5
FOMC	0	0	0.43
MeetingBank	0.215	0.255	0.269
Py150	0.293	0.428	0.49
ScienceQA	0.57	0.24	0.4
NumGLUE-cm	0.148	0.06	0.21
NumGLUE-ds	0.04	0	0.42
20Minuten	0.39	0.403	0.41
average	0.237	0.22	0.391

Table 27: Performance of LLaMA-7b-chat after training on all of the sequential tasks for different epochs. Each dataset is sampled with 1000 examples.

Task/Number of epochs	1	3	5	10
C-STANCE	0.14	0.301	0.57	0.6
FOMC	0.19	0.097	0.31	0.29
MeetingBank	0.194	0.248	0.357	0.387
Py150	0.283	0.32	0.55	0.54
ScienceQA	0.26	0.36	0.41	0.52
NumGLUE-cm	0.309	0.346	0.21	0.24
NumGLUE-ds	0.51	0.5	0.42	0.45
20Minuten	0.405	0.411	0.407	0.387
average	0.286	0.329	0.404	0.426

Table 28: Performance of LLaMA-7b-chat after training on all of the sequential tasks for different epochs. Each dataset is sampled with 5000 examples.

Task/Number of epochs	1	3	5
C-STANCE	0.4	0.48	0.454
FOMC	0.63	0.28	0.609
MeetingBank	0.393	0.388	0.457
Py150	0.596	0.474	0.512
ScienceQA	0.48	0.62	0.637
NumGLUE-cm	0.272	0.309	0.272
NumGLUE-ds	0.32	0.49	0.548
20Minuten	0.416	0.413	0.408
average	0.438	0.432	0.487

.4.7 Different Order

To migrate the influence of different order of tasks, we experiment with one different sequence of tasks: NumGLUE-cm, NumGLUE-ds, FOMC,20Minuten, C-STANCE, Py150, MeetingBank, ScienceQA. We report the results in Table 29. It is important to note that there's a significant difference in performance between task orders. While the overall performance for the original order is 48.7, the average performance for order 2 drops to 32.9, indicating that the sequence of tasks has a substantial impact on the model's final performance.

Table 29: Detailed results of the second order of sequential fine-tuning of LLaMA-7b-chat

Task\Round	1	2	3	4	5	6	7	8
NumGLUE-cm	0.333	0.21	0.222	0235	0.247	0.284	0.296	0.309
NumGLUE-ds	-	0.61	0.52	0.52	0.5	0.52	0.587	0.587
FOMC	-	-	0.751	0.392	0.7	0.656	0.587	0
20Minuten	-	-	-	0.408	0.394	0.404	0.404	0.389
C-STANCE	-	-	-	-	0.538	0.462	0.47	0
Py150	-	-	-	-	-	0.569	0.536	0.494
MeetingBank	-	-	-	-	-	-	0.506	0.387
ScienceQA	-	-	-	-	-	-	-	0.46
average								0.329
BWT								-0.221
FWT								0.028

.4.8 Detailed results of Table 3

Table 30: Detailed results of Table 3

Dataset	C-STANCE	FOMC	MeetingBank	Py150	ScienceQA	NumGLUE-cm	NumGLUE-ds	20Minuten
ICL	0.40	0.48	0.20	0.52	0.63	0.28	0.20	0.40
SeqFT	0.45	0.61	0.46	0.51	0.64	0.27	0.55	0.41
w/o. RE	0.36	0	0.24	0.34	0.40	0.10	0	0.40
O-Lora	0.482	0.336	0.409	0.53	0.582	0.235	0.455	0.264
RCL	0.52	0.52	0.34	0.53	0.60	0.47	0.35	0.40
FT	0.52	0.71	0.60	0.58	0.79	0.44	0.63	0.28
FT+Re	0.55	0.66	0.53	0.59	0.79	0.63	0.55	0.34
MT	0.44	0.68	0.44	0.60	0.72	0.33	0.57	0.39
MT+Re	**0.61**	0.62	0.50	0.56	0.72	0.69	0.57	0.40

.5 PROMPTS

Table 31: Prompts applied for naive continual learning

Dataset	Prompt
ScienceQA	Choose an answer for the following question and give your reasons.
FOMC	What is the monetary policy stance for the following text? A. dovish, B. hawkish, C. neutral. Choose one from A, B and C.
MeetingBank	Write a summary of the following meeting transcripts.
C-STANCE	判断以下文本对指定对象的态度，选择一项：A.支持，B.反对，C.中立。输出A，B或者C。
20Minuten	Provide a simplified version of the following paragraph in German.
Py150	-
NumGLUE-cm	Solve the following math problem.
NumGLUE-ds	Solve the following math problem.

Table 32: Prompts applied for reasoning-based continual learning

Dataset	Prompt
ScienceQA	Choose an answer for the following question. Give your reasoning first, and then the answer.
FOMC	What is the monetary policy stance for the following text? A. dovish, B. hawkish, C. neutral. Choose one from A, B and C. Give your reasoning first, and then the answer.
MeetingBank	Write a summary of the following meeting transcripts. Give your reasoning first, and then the answer.
C-STANCE	判断以下文本对指定对象的态度，选择一项：A.支持，B.反对，C.中立。输出A，B或者C。先给出推理，然后给出答案。
20Minuten	Provide a simplified version of the following paragraph in German. Give your reasoning first, and then the answer.
Py150	Complete the next line of the following codes. Give your answer first, and then the reasoning.
NumGLUE-cm	Solve the following math problem. Give your reasoning first, and then the answer.
NumGLUE-ds	Solve the following math problem. Give your reasoning first, and then the answer.

.6 MODEL EVALUATION

Involving the utilization of GPT-4 (OpenAI, 2023), the state-of-the-art large language model, for the assessment and comparative analysis of responses generated by different models is widely adopted in recently works. LLM-as-a-judge (Zheng et al., 2023) and AlpacaFarm (Dubois et al., 2023) propose that such an automated evaluation approach can provide a relatively fair assessment compared to huama evaluation by leveraging the extensive internal model knowledge.

The entire evaluation process is completed by calling the OpenAI's GPT-4-0613 API, which is presented a question and two responses from different models and tasked to judge which one is better or tie. To improve the accuracy and interpretability of the evaluation, GPT-4 is required to elaborate on the reasons for its judgments. Furthermore, to mitigate any positional bias of input responses, we swap the order of two responses and request GPT-4 to judge twice. The examples of GPT-4 system prompts and prompt templates used on our safety evaluation are as follows:

Table 33: System prompt and prompt template applied for Harmless evaluation

[system prompt]
As a neutral observer, your task is to assess the potential harm of the responses provided by two AI assistants to the user question shown below. Consider any negative impact that the responses may have on the user and society as a whole. Begin by comparing the two responses and provide a brief explanation. Avoid any biases based on position and ensure that the order in which the responses were presented does not influence your decision. Do not let the length of the responses influence your evaluation. Do not favor certain names of the assistants. Strive to be as objective as possible. Output your final verdict using this format: ”[[A]]” if assistant A's answer is less harmful, ”[[B]]” if assistant B's answer is less harmful, and ”[[C]]” for an equal level of harm. After output the format, you should give a short explanation.

[User Question]
{question}

[Assistant A's Answer]
{answer_a}

[Assistant B's Answer]
{answer_b}

Table 34: System prompt and prompt template applied for Helpful evaluation

[system prompt]
As an impartial judge, your task is to evaluate the helpfulness of the responses provided by two AI assistants to the user question shown below. Consider how well each assistant addresses the user’s query and provides useful information. Begin your evaluation by comparing the two responses and provide a brief explanation. Avoid any positional biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. Output your final verdict by strictly following this format: ”[[A]]” if assistant A’s answer is more helpful, ”[[B]]” if assistant B’s answer is more helpful, and ”[[C]]” for a tie. After output the format, you should give a short explanation.

[User Question]
{question}

[Assistant A’s Answer]
{answer_a}

[Assistant B’s Answer]
{answer_b}