TRACE: A Comprehensive Benchmark for Continual Learning in Large Language Models

Anonymous ACL submission

Abstract

Aligned large language models (LLMs) excel in task-solving, instruction-following, and safety. However, their continual learning capabilities have been largely overlooked. Existing continual learning benchmarks lack sufficient challenge for leading aligned LLMs due to their homogeneous task types and low task complexity. To bridge this gap, we present TRACE, a benchmark designed to assess continual learning in LLMs through eight tasks, including domain-specific, multilingual, code generation, and mathematical reasoning. Through systematic experiments on TRACE with six different aligned models ranging from 7B to 70B, we discovered significant declines in both general performance and instruction-following abilities. 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 evolving LLMs to perform well on new tasks while retaining their original capabilities. Our results demonstrate that integrating task-specific cues with meta-rationales significantly reduces catastrophic forgetting and improves task convergence, offering a viable strategy to enhance the adaptability of LLMs in dynamic environments.

1 Introduction

Large Language Models (LLMs) (OpenAI, 2023; Touvron et al., 2023) 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 and 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 and 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 thoroughly researched.

In this paper, we present **TRACE**, a continual learning benchmark designed for aligned LLMs. Our benchmark consists of eight distinct datasets spanning challenging tasks including domain-specific tasks, multilingual capabilities, code generation, and mathematical reasoning. 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 abili-**

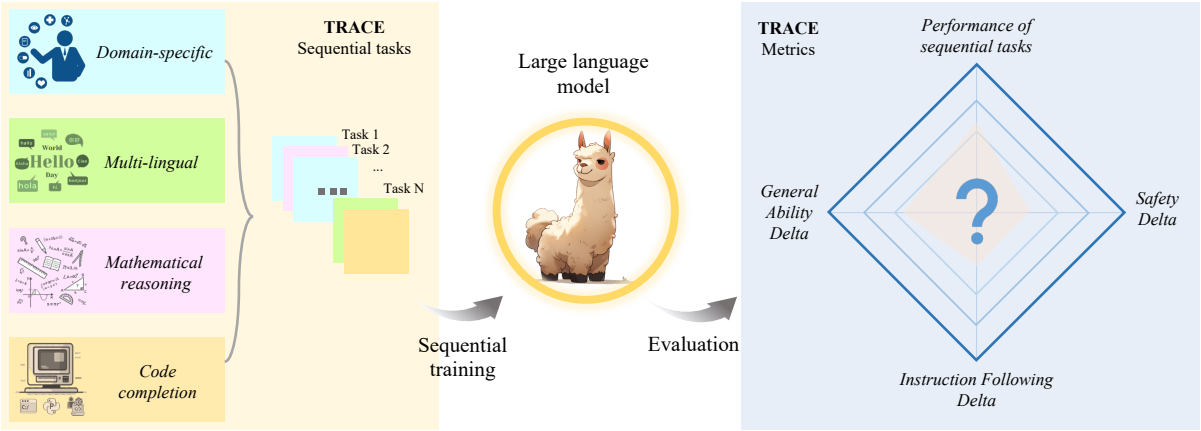


Figure 1: An overview of TRACE benchmark. TRACE consists of two main components: 1) A selection of eight datasets constituting a tailored set of tasks for continual learning, covering challenges in domain-specific tasks, multilingual capabilities, code generation, and mathematical reasoning. 2) A post-training 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.

ties and instruction-following capabilities after training on TRACE. For instance, the accuracy of llama2-chat 13B on the gsm8k dataset dropped from 43.14% to merely 2.12%. 2) Our safety assessment revealed that **LLMs trained on TRACE maintained their safety**. This indicates that companies offering fine-tuning APIs, such as OpenAI, can ensure model safety by filtering out malicious data, thereby mitigating concerns about catastrophic forgetting due to continual learning. 3) **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. 4) 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.

Regrettably, our experiments demonstrate that some traditional continual learning methods (Kirkpatrick et al., 2017; Farajtabar et al., 2020) do not work well in the context of LLMs. 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 and 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.

2.2 CL Benchmarks in NLP

The most recognized CL benchmark for NLP encompasses five text classification datasets (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). 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.

3 Preliminaries

In supervised continual learning setting (Ke and Liu, 2022; Wang et al., 2023a), a sequence of tasks $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$ arrive in a streaming fashion. Each task $\mathcal{D}_t = \left\{\left(\boldsymbol{x}_i^t, y_i^t\right)\right\}_{i=1}^{n_t}$ 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 and Ranzato, 2017)), and backward transfer (BWT (Lopez-Paz and 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

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 six 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.

Dataset	Avg len	Lang	Count
Domain-specific			
ScienceQA	210	English	5,000
FOMC	51	English	5,000
MeetingBank	2853	English	5,000
Multi-lingual			
C-STANCE	127	Chinese	5,000
20Minuten	382	Germany	5,000
Code completion			
Py150	422	Python	5,000
Mathematical reasoning			
NumGLUE-cm	32	English	5,000
NumGLUE-ds	21	English	5,000

Table 1: An overview of dataset statistics in TRACE. 'Avg len' represents the word count for English, German, and code datasets, and the character count for Chinese.

According to these principles, 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 1, we get 40,000 training examples and 16,000 testing examples in total.

Domain-Specific. ScienceQA (Lu et al., 2022) is a multi-hop QA dataset collected from elementary and high school science curricula, with a rich domain diversity from natural science, social science, and language science, requiring the model of reasoning ability and science knowledge. In TRACE, we include only non-multimodal examples. FOMC (Shah et al., 2023) is a hawkish-dovish classification task, which is novel in the financial domain. The dataset is divided into three subsets: data on meeting minutes, press conference data, and speech data. We use a combination of them. MeetingBank (Hu et al., 2023) is a new benchmark dataset for city council meeting summarization, an unstudied domain. It demands a global understanding of the whole long context.

Multi-lingual. The cross-lingual abilities of LLMs are limited by their vocabulary and pre-training corpus. For instance, LLaMA's vocabulary contains few Chinese tokens, affecting its Chinese processing abilities. C-STANCE (Zhao et al., 2023) is the first Chinese dataset for zero-shot stance detection collected from Sina Weibo, one of the most popular Chinese social media sites. It includes two challenging subtasks:

target-based stance detection and domain-based stance detection. In TRACE, we include the target-based one, which means the targets in testing examples are unseen during training. 20Minuten (Rios et al., 2021) is a text simplification dataset that pairs full articles with simplified summaries from a Swiss news magazine. We utilize this dataset to assess the ability in German.

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), which requires the model to generate the next line given the lengthy code input. The corpus Py150 (Lu et al., 2021) contains 150,000 Python programs collected from GitHub repositories. Since the golden labels of the testing dataset are not available by (Lu et al., 2021), we randomly divide each Python code in Py150 into two parts, taking the first part as inputs and the next line as labels.

Mathematical reasoning. Mathematical problems are always used to evaluate the reasoning ability of models. NumGLUE (Mishra et al., 2022) is an 8-task benchmark far from solved including state-of-the-art large-scale language models performing significantly worse than humans. We include the first two tasks of NumGLUE, both requiring arithmetic reasoning. It is worth noting that both datasets have original labels consisting only of numbers, without associated inference processes. The first one evaluates the common sense of models, while the second one requires some grade-school scientific knowledge.

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. In the TRACE benchmark, we incorporate three metrics tailored to evaluate distinct dimensions of model performance following training: **General Ability Delta**, **Instruction Following Delta**, and **Safety Delta**. Each metric is designed to capture shifts in different areas—general abilities, instruction-following, and safety, respectively. Despite their targeted focuses, all three metrics are calculated using a consistent formula:

$$\Delta R_t^X = \frac{1}{N}\sum_{i=1}^{N}(R_{t,i}^X - R_{0,i}^X)$$

where X represents the specific metric focus—general ability, instruction-following, or safety—and N is the number of tasks. This unified formula underscores the systematic approach TRACE adopts to evaluate different dimensions of LLM performance enhancements or declines post-training.

In the development of evaluation metrics, integrating existing and newly introduced indicators into a single, unified metric might appear straightforward through a weighted summary. However, the implementation of such a composite metric necessitates careful prioritization of the capabilities most valued by users. Therefore, this paper does not propose a universal metric for ranking models.

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 different approaches:

Sequential Full-Parameter Fine-Tuning (SeqFT) It trains all model parameters in sequence.

LoRA-based Sequential Fine-Tuning (LoraSeqFT) Following (Hu et al., 2021), we only fine-tuned the low-rank LoRA matrices and leave the LLM backbone fixed. This method is chosen based on prior findings of reduced forgetting with "Efficient Tuning" (Liu and Huang, 2023).

Replay-based Sequential Fine-Tuning (Replay) We incorporate data from LIMA along with 10% of data from previous tasks into our training dataset.

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.

	SeqFT			LoRASeqFT			Replay		
	OP	**BWT**	**FWT**	**OP**	**BWT**	**FWT**	**OP**	**BWT**	**FWT**
LLaMA2-7B	48.7	−8.3%	2.4%	12.7	−45.7%	0.8%	55.5	**2.6%**	−0.2%
LLaMA2-13B	49.9	−7.0%	**2.5%**	28.0	−36.5%	1.5%	56.5	0.4%	2.2%
LLaMA2-70B	-	-	-	**45.2**	−17.5%	**1.7%**	56.6	0.4%	1.9%
Vicuna-7B	49.2	−8.4%	1.7%	33.4	−23.7%	0.9%	55.3	0.2%	0.5%
Vicuna-13B	**51.2**	**-5.9%**	2.1%	31.6	−28.4%	1.1%	**56.9**	0.6%	**2.4%**
Baichuan2-7B	43.4	−15.4%	1.8%	43.8	**-9.0%**	1.0%	51.7	1.1%	−4.0%
	EWC			**OGD**			**PP**		
	OP	**BWT**	**FWT**	**OP**	**BWT**	**FWT**	**OP**	**BWT**	**FWT**
LLaMA2-7B	32.3	−20.6%	1.0%	29.7	−22.6%	0.9%	49.2	−0.3%	1.9%
LLaMA2-13B	35.7	−19.7%	1.1%	36.6	−18.6%	0.7%	51.3	−1.1%	**2.1%**
LLaMA2-70B	-	-	-	-	-	-	**54.6**	−0.4%	1.7%
Vicuna-7B	37.2	−16.8%	**1.3%**	39.9	−15.7%	**1.6%**	50.4	**0.2%**	1.6%
Vicuna-13B	36.5	−18.6%	1.1%	40.4	−16.7%	0.6%	53.6	−0.6%	2.0%
Baichuan2-7B	**44.5**	**-7.6%**	0.8%	**44.7**	**-7.6%**	1.1%	46.6	−1.7%	0.9%
	L2P			**LFPT5**			**O-Lora**		
	OP	**BWT**	**FWT**	**OP**	**BWT**	**FWT**	**OP**	**BWT**	**FWT**
LLaMA2-7B	43.7	−9.5%	**2.6%**	49.2	−6.7%	1.5%	41.3	−6.2%	**1.6%**
LLaMA2-13B	46.4	−6.4%	1.6%	50.4	**-5.6%**	1.1%	43.7	−4.4%	0.7%
LLaMA2-70B	**53.2**	**-1.6%**	1.4%	-	-	-	47.7	−5.7%	0.4%
Vicuna-7B	48.6	−9.2%	1.5%	**51.6**	−7.8%	0.7%	42.6	−6.8%	1.1%
Vicuna-13B	52.4	−4.3%	2.2%	51.2	−6.6%	**1.9%**	**47.9**	**-3.2%**	0.9%
Baichuan2-7B	47.6	−8.6%	0.6%	48.3	−6.7%	1.7%	45.6	−5.4%	1.2%

Table 2: Overall Performance (OP), Forward Transfer (FWT), and Backward Transfer (BWT) for 6 baseline models and 9 baseline methods.

Other Continual Learning Baselines We also demonstrate the experiment results on a few continual learning baselines, including EWC (Kirkpatrick et al., 2017), ODG (Farajtabar et al., 2020), PP (Razdaibiedina et al., 2023), L2P (Wang et al., 2022d), LFPT5 (Qin and Joty, 2022). Given that EWC and OGD require storing parameters or gradients from past tasks, which is impractical for the full parameter setting of LLMs, we validate these methods using LoRA.

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), 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., 2022c), comprising 175 user-oriented prompts, and the LIMA dataset (Zhou et al., 2023), which assembles 300 prompts from community Q&A and manual examples.

To evaluate *safety* changes, we use the CoNa dataset (Bianchi et al., 2023), which consists of 178 expert-annotated samples focused on instructions related to hateful speech generation.

4.3.3 Implementation Details

In the training phase, we trained models with and without LoRA adapters using 5000 samples at learning rates of 1e-5 and 1e-4, respectively. 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. All general benchmark evaluations were conducted using the Open-Compass toolkit (Contributors, 2023), adopting its default configuration. The detailed settings can be found in Appendix A.

	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

Table 3: Comparison of the general abilities. blue means increase, while red means decrease.

4.4 Main Results

We conducted experiments with various task orders. Results for the default order are presented in the main text, while results for additional orders are detailed in Appendix D.7.

4.4.1 Performance of Target Sequential Tasks

Table 2 showcases the performance of six 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%). This suggests that when the focus is primarily on sequential tasks, full parameter fine-tuning should be prioritized over parameter-efficient methods like LoRA.

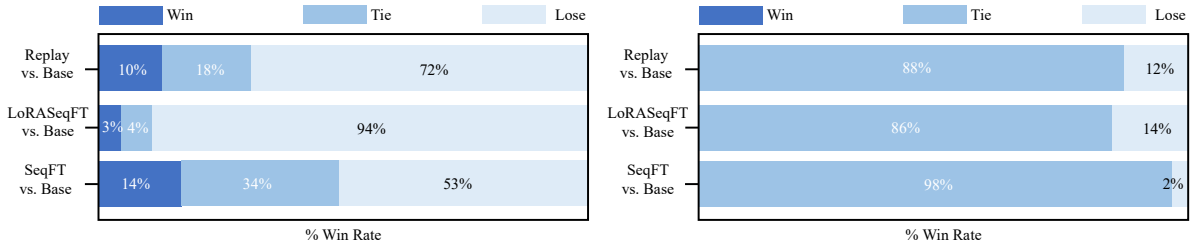
Catastrophic Forgetting in LLaMA-2-70B-Chat Despite achieving a high overall performance (OP) score of 45.2, the llama2-70B-Chat model exhibits significant catastrophic forgetting, as evidenced by its BWT of -17.5%. This substantial decline in retaining previously learned information indicates a vulnerability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

4.4.2 Variation of General Ability

Table 3 presents the evaluations of various LLM models concerning general abilities. The degree of general ability forgetting in LLMs can be analyzed from three perspectives.

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, all models show a noticeable decline in math and reasoning abilities, highlighting their sensitivity to new task learning.



(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.

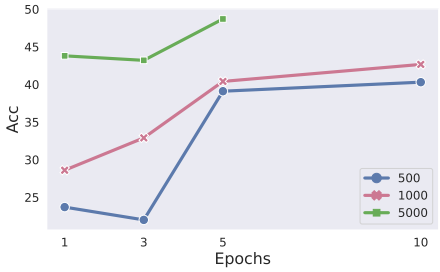


Figure 3: Performance evaluation of LLaMA-2-7B-Chat's SeqFT on TRACE benchmark across varying sample sizes (500, 1000, 5000) and training epochs (1, 3, 5, 10 (except for 5000)).

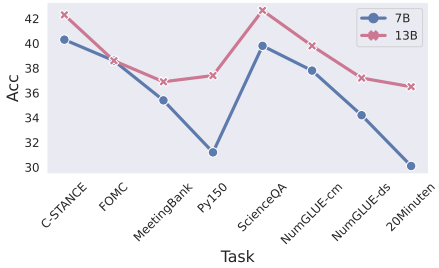


Figure 4: Evolution of LLMs' reasoning capabilities post-training, measured using the BBH metric. Results for LLaMA-2-7B-chat and LLaMA-2-13B-chat are reported.

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. Intriguingly, other languages on TyDiQA also show enhancements and declines, suggesting potential cross-linguistic transfer effects. **4)** Performance shifts on PIQA for most models are subtle, indicating the relative robustness of commonsense knowledge during continual learning.

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 LLMs based on 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. All three training methods show a significant decline in instruction-following capabilities, with the most pronounced decline 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.

4.5 Ablation Study

Data Quantity & Training Steps. Figure 3 shows the impact of data volumes and training steps on target task performance. For LLaMA-2-7B-Chat's SeqFT, we trained with 500, 1000, and 5000 samples from each dataset over 1, 3, 5, 10 epochs, respectively. The results indicate that performance improves with an increase in data volume, reaching optimal at 5000 samples. Furthermore, performance does not increase after 5 epochs, indicating a balance for task optimization and capabilities retention.

Decoupling the Impact of Different Tasks. Results from section 4.4.2 reveal a notable decline in reasoning and mathematical skills after training on our benchmark. 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. Interestingly, the model's reasoning skill improves after the ScienceQA task but declines for others. Unlike NumGLUE tasks, which lack clear reasoning paths, ScienceQA provides explicit reasoning paths in its answers, indicating that including reasoning paths in training maybe benefit the model's reasoning skills.

4.6 Reasoning-augmented Continual learning

Why do LLMs underperform on TRACE, both in target tasks and maintaining inherent capabilities? Two likely reasons: First, neural networks tend to overfit the specific output formats of new tasks (Wang et al., 2022b). Second, as most tasks involve direct result prediction, LLMs may learn shortcuts (Wang et al., 2022a) within the training data, limiting their ability to generalize to new data.

With these insights and our experimental findings in 4.5, we propose the Reasoning-augmented Continual Learning (RCL). As depicted in Figure 5, RCL involves two phases: automated reasoning annotation and sequential training on the augmented datasets. GPT-4 generates reasoning paths for all entries, based on prompts designed by domain experts for each task. These are then validated against the ground truth. Manual inspection of a 100-sample subset results in a 94% approval rate, demonstrating GPT-4's reliability. Following this, supervised training is conducted, keeping hyperparameter settings consistent with baselines.

4.6.1 Performance on Sequential Tasks

Table 4 provides comparisons of the performance of RCL against other baselines. 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 consistently boosts performance over the original dataset. Our approach yields results comparable to the SeqFT method using only 500 samples instead of 5000. By utilizing fewer datasets and training steps, our method also helps preserve the inherent capabilities of LLMs more effectively.

Methods	OP	BWT	FWT
EWC	32.3	−20.6%	1.0%
OGD	29.7	−22.6%	0.9%
L2P	43.7	−9.5%	2.6%
LFPT5	49.2	−6.7%	1.5%
PP	49.2	**-0.3%**	1.9%
O-Lora	41.3	−6.2%	1.6%
ICL	39.5	-	-
ICL+Re	41.1	-	-
SeqFT(0.5k)	23.0	−19%	0.3%
RCL(0.5k)	46.6	−13%	2.7%
SeqFT(5k)	48.7	−8.3%	2.4%
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	-	-

Table 4: Comparison of RCL with different baselines. **Single FT** refers to fine-tuning the model on a single task, **Re** refers to reasoning-augmented, and **MT** refers to Multi-task training.

4.6.2 Original Performance Retention

Impacts on General Ability Figure 6 shows RCL's performance on general abilities matches SeqFT and Replay. Yet, RCL excels in reasoning tasks like GSM and BBH, where it surpasses SeqFT and Replay by 12.7 and 13.2 points, respectively. This highlights RCL's effectiveness in enhancing reasoning skills through reasoning paths. Additionally, combining RCL with replay boosts its reasoning task performance.

Impacts on Instruction-Following 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.

5 Conclusion

To address the insufficiencies of existing CL benchmarks, we introduce TRACE, a comprehensive benchmark with eight challenging tasks and well-rounded metrics. Our experiments show that for LLMs, catastrophic forgetting remains, and a clear drop in general abilities is observed during continual learning. Besides, our RCL method highlights the importance of using reasoning in training while alleviating the above phenomena. We believe this area is crucial and hope our work serves as a foundation for future research.

6 Limitations

While our proposed TRACE benchmark covers a variety of tasks, expanding it to include real-time interaction and multi-modal integration could provide a more comprehensive assessment of LLM capabilities. Additionally, future versions should aim to minimize any potential biases introduced by the current task selection or dataset composition, ensuring a more balanced and comprehensive representation of tasks that better reflect diverse real-world applications.

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. 2021. Program synthesis with large language models. *ArXiv*, abs/2108.07732.

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. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *Preprint*, arXiv:2308.14508.

Baichuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

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

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

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. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

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

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. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

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

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. 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.

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

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

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

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

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

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

Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. Meetingbank: A benchmark dataset for meeting summarization. *Preprint*, arXiv:2305.17529.

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

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

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

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

Lei Liu and J. Huang. 2023. 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*.

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

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Preprint*, arXiv:2209.09513.

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. 2021. Codexglue: A machine learning benchmark dataset for code understanding and generation. *Preprint*, arXiv:2102.04664.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, A. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Annual Meeting of the Association for Computational Linguistics*.

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

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks. *Preprint*, arXiv:2204.05660.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Chengwei Qin and Shafiq Joty. 2022. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. *Preprint*, arXiv:2110.07298.

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. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *ArXiv*, abs/2307.16789.

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

Annette Rios, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. 2021. A new dataset and efficient baselines for document-level text simplification in German. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 152–161, Online and in Dominican Republic. Association for Computational Linguistics.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *Conference on Empirical Methods in Natural Language Processing*.

Agam Shah, Suvan Paturi, and Sudheer Chava. 2023. Trillion dollar words: A new financial dataset, task & market analysis. *Preprint*, arXiv:2305.07972.

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

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

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. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A

stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

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

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

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. *Preprint*, arXiv:2310.14152.

Xiao Wang, Shihan Dou, Limao Xiong, Yicheng Zou, Qi Zhang, Tao Gui, Liang Qiao, Zhanzhan Cheng, and Xuan-Jing Huang. 2022a. Miner: Improving out-of-vocabulary named entity recognition from an information theoretic perspective. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5590–5600.

Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S Dhillon, and Sanjiv Kumar. 2022b. Preserving in-context learning ability in large language model fine-tuning. *arXiv preprint arXiv:2211.00635*.

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

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

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022d. Learning to prompt for continual learning. *Preprint*, arXiv:2112.08654.

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

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Chenye Zhao, Yingjie Li, and Cornelia Caragea. 2023. 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)*, pages 13369–13385, Toronto, Canada. Association for Computational Linguistics.

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

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

Appendices

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

B Trace dataset statistics

In this section, we represent the overview of dataset statistics, including source, average length, metric, language, and number of samples of each dataset in the TRACE benchmark. We demonstrate the dataset details in Table 1.

C Reasoning-based Continual Learning

Figure 5 shows an overview of our reasoning-augmented continual learning method. Table 5 and Figure 6 shows the results of RCL in preserving the

instruction-following abilities and general abilities, respectively.

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

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

D 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, LlaMA2-70b-chat, Vicuna-7b and Vicuna-13b. The results are shown in Table 6 - 25.

D.1 In-Context Learning

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

D.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.

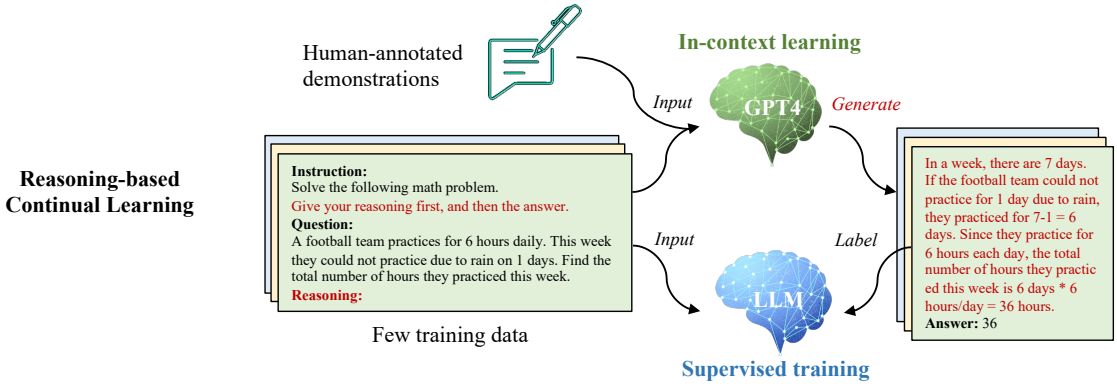


Figure 5: An overview of Reasoning-augmented continual learning. The strategy involves two stages: 1) Automatically annotating sample reasoning paths using GPT-4. 2) Continual learning on augmented datasets.

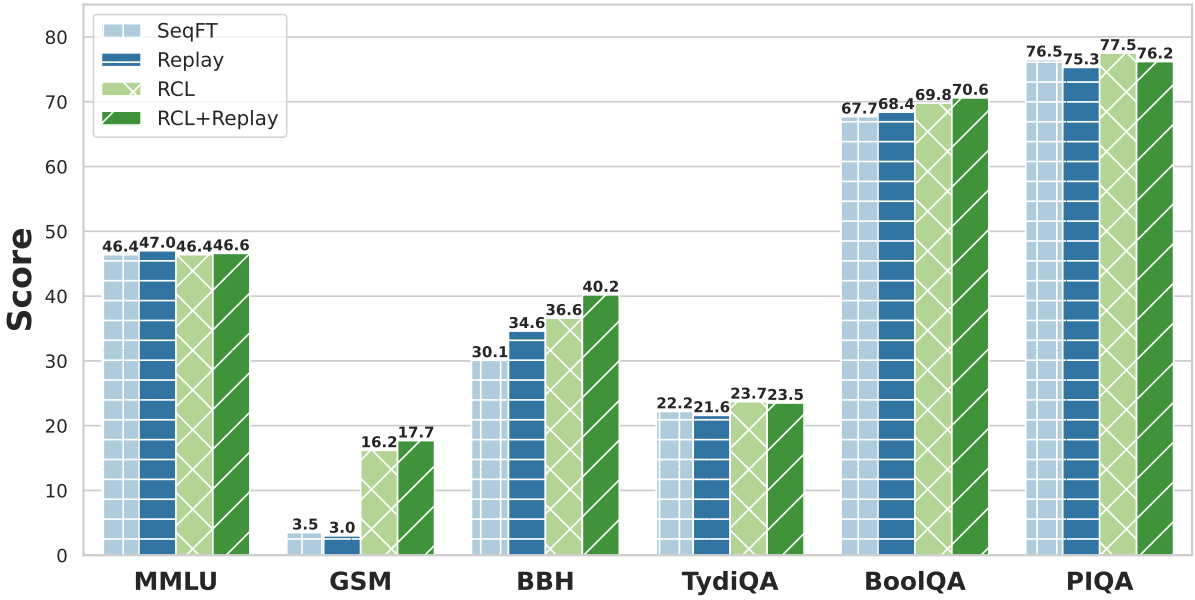


Figure 6: OpenCompass Evaluation Results. **RCL+Replay** refers to combining our RCL method with replay method. The model we use is LLaMA-2-7b-Chat.

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

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

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

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 8: Detailed results of sequential fine-tuning of LLaMA-7b-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 9: Detailed results of sequential fine-tuning of LLaMA-13b-chat.

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

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

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

D.3 SeqLoraFT method

Table 13 - 18 shows the detailed performance of different models of each round during the continual learning. SeqLoRAFT represents sequential fine-tuning 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 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	0.511	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 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	0.62	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 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	0.53	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 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	0.514	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 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	0.524	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

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

D.4 Replay method

Table 19 - 24 shows the detailed performance of different models of each round during the continual learning 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 19: Detailed results of continual learning of Baichuan-7b with replay data.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.471	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 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	0.5	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 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	0.519	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 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	0.5	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 23: Detailed results of continual learning of Vicuna-7b with replay data.

Task\Round	1	2	3	4	5	6	7	8
C-STANCE	0.56	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

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

D.5 RCL method

Table 25 shows the detailed performance of LLaMA2-7b-chat of each round during the continual learning. RCL represents reasoning-based continual learning.

D.6 Different amounts of data and training steps

Table 26 - 28 shows the performance of LLaMA2-7b-chat with different numbers of data and training epochs.

D.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.

However, our benchmark, TRACE, diverges from traditional benchmarks by featuring a diverse array of datasets and metrics. This diversity is inherent to the nature of continual learning tasks, rather than a limitation of our benchmark design. Current validation methods for Large Language Models (LLMs) across various capabilities also do not rely on a single metric or dataset. Similar to how LLMs are assessed on multiple benchmarks to validate different capabilities, TRACE's comprehensive approach is essential for evaluating continual learning comprehensively. Hence, the necessity for varied configurations in TRACE is dictated by the multifaceted nature of continual learning itself, not by a flaw in our methodology.

D.8 Detailed results of Table 4

We report the detailed experimental results of Table 4 in Table 30.

D.9 Model Selection

Despite the inherent differences among models, it is feasible for users to make informed decisions based on their specific requirements. For instance, if a user prioritizes the preservation of general capabilities, Vicuna-7b-v1.5 emerges as a suitable choice. Furthermore, users have the flexibility to weight all the metrics and prioritize metrics that more align with their objectives, enabling them to select the model that best fits their needs.

E Prompts

We show the prompts used across our experiments. The prompts of our naive sequence training are shown in Table 31. While for RCL, the prompts are shown in Table 32.

F 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 recent 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 human evaluation by leveraging the extensive internal model knowledge.

The entire evaluation process is completed by calling the OpenAI's GPT-4-0613 API, which presents a question and two responses from different models and tasks 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 shown in Table 33 and Table 34.

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

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

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 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	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 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
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

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\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

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

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

Table 30: Detailed results of Table 4

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 31: Prompts applied for naive 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.

Table 32: Prompts applied for reasoning-based continual learning

[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 33: System prompt and prompt template applied for Harmless 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}

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