Exploring Continual Fine-Tuning for Enhancing Language Ability in Large Language Models

Divyanshu Aggarwal[1]* **Sankarshan Damle**[2]* **Navin Goyal**[1] **Satya Lokam**[1] **Sunayana Sitaram**[1]
[1]Microsoft Research India [2]LIA, EPFL
t-daggarwal@microsoft.com sankarshan.damle@epfl.ch
{navingo,satya.lokam,sunayana.sitaram}@microsoft.com

Abstract

A common challenge towards the adaptability of Large Language Models (LLMs) is their ability to learn new languages over time without hampering the model's performance on languages in which the model is already proficient (usually English). Continual fine-tuning (CFT) is the process of sequentially fine-tuning an LLM to enable the model to adapt to downstream tasks with varying data distributions and time shifts. This paper focuses on the language adaptability of LLMs through CFT. We study a two-phase CFT process in which an English-only end-to-end fine-tuned LLM from Phase 1 (predominantly Task Ability) is sequentially fine-tuned on a multilingual dataset – comprising task data in new languages – in Phase 2 (predominantly Language Ability). We observe that the "similarity" of Phase 2 tasks with Phase 1 determines the LLM's adaptability. For similar phase-wise datasets, the LLM after Phase 2 does not show deterioration in task ability. In contrast, when the phase-wise datasets are not similar, the LLM's task ability deteriorates. We test our hypothesis on the open-source MISTRAL-7B and LLAMA-3-8B models with multiple phase-wise dataset pairs. To address the deterioration, we analyze tailored variants of two CFT methods: layer freezing and generative replay. Our findings demonstrate their effectiveness in enhancing the language ability of LLMs while preserving task performance, in comparison to relevant baselines.

1 Introduction

With ever-increasing adoption of LLMs in real world applications and expanding multilingual user bases of these applications, it is important to cater these models to wide enough multilingual audiences. Model training is compute hungry, and there is an abundance of both labelled and unlabelled data in English as compared to other languages [36]. As such, it is imperative to find efficient ways to use pre-trained or fine-tuned models to improve performance on other languages. In this paper, we refer to a model's ability in non-English languages as predominantly its *language ability* (LA), which can be achieved without relying on large amounts of data in those languages. Instead, we can exploit the predominantly *task ability* (TA) learned from English data.

To this end, researchers use techniques like continual pre-training, continual fine-tuning or language adaption to adapt models to a newer set of languages to enhance their language abilities (refer to §2). While these techniques are effective, they are highly task-specific. Furthermore, existing techniques for multilingual LLMs rely on parallel data, old fine-tuning data, or old and new set of parameters. Parameter efficient techniques like LoRA [20] are also widely used to efficiently fine-tune LLMs on multilingual data. However, such techniques show both: *catastrophic forgetting* on English and incapability to exploit the task ability that the model receives from the English fine-tuning data [1].

*Equal Contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

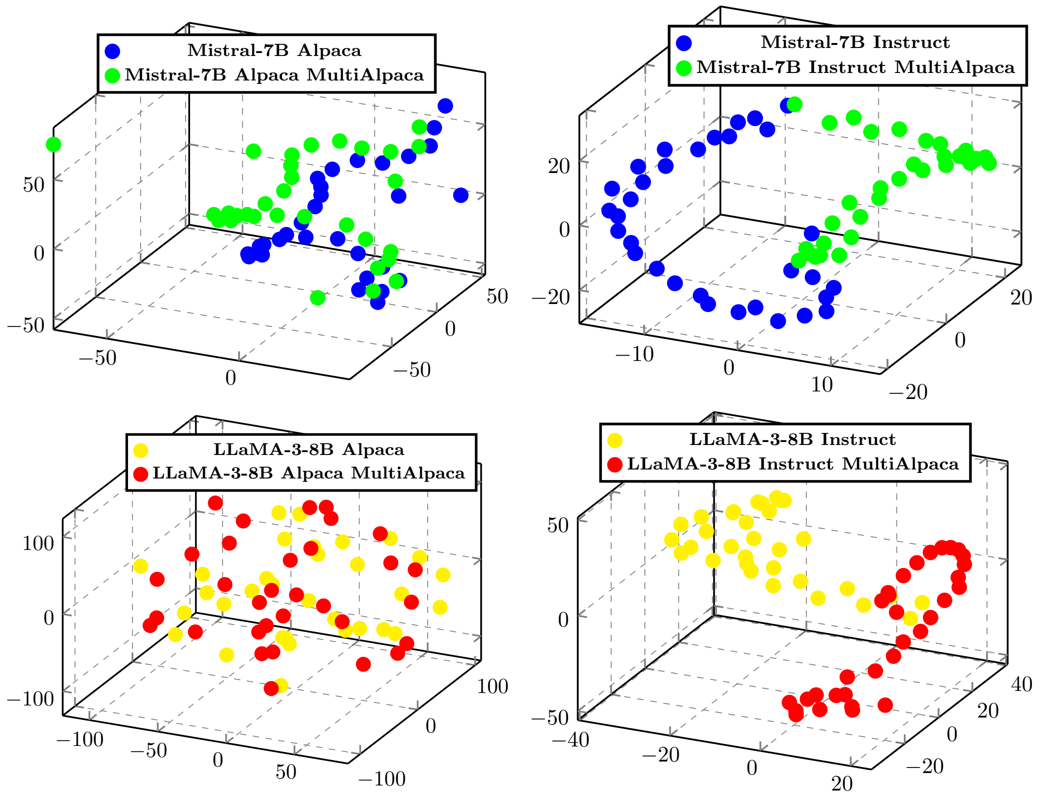

Figure 1: Comparing hidden activations for MISTRAL-7B and LLAMA-3-8B during our two-phase continual fine-tuning process. We prompt each model with examples from MTBENCH [48], and visualize the similarity between the mean hidden activations, for each model layer. For datasets that encode "similar" tasks (ALPACA & MULTIALPACA), model's task ability does not decline (e.g., 3% gain for `IFEval`). For non-similar datasets (Instruct & MULTIALPACA), the task ability declines (e.g., 8% decline for `IFEval`). Here, Phase 2 model representations do not align with Phase 1's; thus, suggesting greater model weight interference and a decline in task ability.

In such a setting, we want to enhance the model's language ability (other than English) while preserving the task ability achieved via (firstly) English fine-tuning. This setting results in the challenge of catastrophic forgetting, i.e., the model's task ability on English may decline while fine-tuning on multilingual data [31]. Furthermore, a trivial solution that fine-tunes on the mixture of multilingual and English-only data may be sub-par (e.g., due to language relatedness [11]). Hence, it is challenging to improve an LLM's language ability while preserving its performance on English.

Our Approach. We use a two-phase continual fine-tuning (CFT) technique for language adaption. We study the effects of various English and multilingual instruction tuning datasets when the models are fine-tuned in two phases: where Phase 1 is fine-tuning the model in English to improve its task ability and then fine-tuning it on a proportionally-sized multilingual dataset in Phase 2. In Phase 1, we use ALPACA [41] and OPENORCA [29], and in Phase 2 we use MULTIALPACA [43] and MOPENORCA (§4.1).

We perform this study on two open-source models, namely LLAMA-3-8B and MISTRAL-7B. We also use fine-tuned versions of them, namely LLAMA-3-8B-INSTRUCT and MISTRAL-7B-INSTRUCT, as off-the-shelf Phase 1 fine-tuned models. We quantify a model's task ability based on its performance on four English datasets: (i) two for instruction following (i.e., `IFEval` [49] and `Alpaca Eval` [28]) and (ii) two for reasoning tasks (i.e., MMLU [19] and `HellaSwag` [45]). Likewise, we quantify a model's language ability based on its performance on (i) two question answering tasks (i.e., `MLQA` [27] and `XQuAD` [3]) and (ii) `XLSUM` [17], a summarization task.

Our Contributions. First, we observe that when phase-wise English and multilingual datasets encode different tasks, we see a decline in the Phase 2 model's performance on English. On the other hand, when Phase 1 and Phase 2 datasets encode similar tasks, the Phase 2 model's performance on English improves (refer to Figure 1). Second, to quantify the similarity of these phase-wise datasets, we introduce two metrics based on language-agnostic embeddings and model representations. We show that our quantification correlates with the decline in task ability (§4.3). Third, we study the efficacy of two tailored variants of existing CFT strategies to mitigate the decline in task ability after Phase 2 fine-tuning, while also boosting the language ability. The first strategy we study is *generative replay*, i.e., using instructions from a similar English counterpart of the Phase 2 dataset to generate replay data using the Phase 1 model. The second strategy uses heuristic-based *layer freezing*. Here, we use the weight difference between the Base and Phase 1 models to pick specific layers for freezing during Phase 2 fine-tuning. We study the gains in task and language ability of these strategies compared to specific baselines (§5).

2 Related Work

Continual Learning in LLMs. In general, continual learning in LLMs can be broadly categorized into (i) continual pre-training (CPT) and (ii) continual fine-tuning (CFT). In CPT, the LLMs are continuously pre-trained to adapt to new domains or tasks by continuously updating them with new data alongside the existing data [37]. CPT builds on the existing LLM's knowledge and is more computationally efficient than retraining an LLM using the current and old pre-training data [16]. CPT is employed when distributional shifts occur (i) over time [2, 21, 22], (ii) across languages [24, 14, 5] or (iii) across domains [25, 15, 44].

On the other hand, CFT involves training the LLM on successive downstream tasks with varying data distribution or time shifts [37]. CFT comprises fine-tuning for different tasks [7], instruction-tuning [6], model refinement/editing [46] and alignment [40]. Recent literature also focuses on using CFT to assist the LLM to learn new languages [35, 34, 4].

CFT: Enhancing LLMs Multilingual Abilities. Cahyawijaya et al. [6] propose InstructAlign which uses cross-lingual alignment and episodic replay to align an LLM's pre-trained languages to unseen languages but requires parallel data and previous task data. Shaham et al. [36] introduces multilinguality during the first instruction fine-tuning phase which improves an LLM's instruction following capability across languages. He et al. [18] show catastrophic forgetting during CFT and use techniques such as joint fine-tuning and model regularization to mitigate it. However, these techniques are computationally expensive or require access to previous task data.

Language Adaption. This set of works looks at language and task adaption by adjusting the model to understand new languages and enhancing its performance on specific tasks through fine-tuning, respectively [9, 47, 33]. For instance, Chen et al. [9] perform task adaption by fine-tuning the model on downstream task data. For language adaption, they fine-tune only the token embedding layer, helping the model learn specific lexical meanings of new languages. Language and task ability are either trained in parallel or sequentially. However, in this paper, we try to incorporate language ability in models with the constraint that they may have already learned task ability (e.g., MISTRAL-7B-INSTRUCT). To the best of our knowledge, this is a first attempt at studying the effect of task and language self-instruct datasets on an LLM's multilingual ability through CFT.

3 Enhancing Language Ability through Continual Fine-tuning

A common recipe to training LLMs to learn new languages is to use a training paradigm that focuses on *task* and *language* adaption [9]. Concretely, we define task adaption as the model's ability to comprehend the input text and then provide a suitable output. We refer to language adaption as the model's ability to perform those tasks in languages other than English.

In Task Adaption, the LLM is trained to follow instructions, usually using labeled English data. Language Adaption focuses on training the LLM to understand text from newer languages. Task adaptation leverages cross-lingual transfer, facilitating language adaptation to a certain degree. However, this process can result in a decline in the LLM's task adaptation performance due to the

Model	Phase 1 (P1) Dataset	Phase 2 (P2) Dataset	IFEval (↑) P1	IFEval (↑) P2	Alpaca Eval (↑) P1	Alpaca Eval (↑) P2	MMLU (↑) P1	MMLU (↑) P2	HellaSwag (↑) P1	HellaSwag (↑) P2	Average P1	Average P2
MISTRAL-7B	ALPACA	MULTIALPACA	0.364	0.395	0.12	0.16	0.552	0.573	0.581	0.616	0.404	0.436
	Instruct		0.550	0.462	0.35	0.15	0.575	0.533	0.641	0.416	0.529	0.390
LLAMA-3-8B	ALPACA		0.277	0.326	0.10	0.11	0.231	0.242	0.556	0.567	0.291	0.311
	Instruct		0.735	0.182	0.14	0.10	0.340	0.239	0.533	0.278	0.437	0.2

Table 1: Task Ability results for two-phase Continual Fine-tuning (CFT). When the phase-wise datasets are similar (Definition 1 and Definition 2), task ability post Phase 2 (P2) fine-tuning *consistently* improves (denoted with green). When the phase-wise datasets are not similar, we see a *significant* decline in task ability post Phase 2 (P2) fine-tuning (denote with red).

risk of catastrophic forgetting. Despite this challenge, task adaptation often yields greater benefits compared to relying solely on cross-lingual transfer for language adaptation.

Continual Fine-tuning for Language Adaption. To improve the language adaption of LLMs, we re-imagine the above recipe as a two-phase CFT process. We have:

- Phase 1. We fine-tune a base LLM end-to-end on an English instruction dataset. Phase 1 aims to predominantly teach the LLM instruction following ability, which we refer to as *task ability*.
- Phase 2. Here, we use the fine-tuned LLM from Phase 1 and further end-to-end fine-tune it on a Multilingual instruction dataset. Unlike Chen et al. [9], in our setting, the data in Phase 2 is labeled. However, compared to Phase 1, Phase 2's dataset is geared towards enhancing the LLM's *language ability*, and comprises multiple languages with fewer data points per language.

This paper relates English fine-tuning with task ability enhancement as English fine-tuning predominantly helps in the task ability of LLMs. Whereas multilingual fine-tuning predominantly helps with an LLM's language ability.

CFT for Language Adaption: Challenges. The primary challenge in our two-phase fine-tuning process is that the LLM's language ability must not come at the cost of its task ability. We impose two additional constraints based on real-world scenarios. First, in Phase 2, we cannot re-use Phase 1's dataset. Often instruction fine-tuned LLMs are available without their corresponding datasets (e.g., MISTRAL-7B-INSTRUCT [23]). Second, in Phase 2, we cannot use the weights of the Phase 1 model during training, as saving both old and new set of parameters on the GPU for training would be computationally expensive.

In a nutshell, we focus on CFT for language adaption for an LLM while preserving the model's task ability.

4 Evaluating Task & Language Ability for Multilingual CFT

4.1 Experiment Setup & Evaluation Tasks

Fine-tuning Models. We continually fine-tune open-source MISTRAL-7B [23] and LLAMA-3-8B [12] LLMs for language adaption.

Fine-tuning Datasets. For our phase-wise datasets, we use the open-source ALPACA [41], MULTIALPACA [43], and OPENORCA [29] datasets. ALPACA is a self-instruct English-only dataset. MULTIALPACA is a multilingual dataset created by translating ALPACA's seed tasks to 11 languages and using GPT-3.5-Turbo for response collection. The languages are in equal proportions and are "French", "Arabic", "German", "Spanish", "Indonesian", "Japanese", "Korean", "Portuguese", "Russian", "Thai", and "Vietnamese". The appendix (§**??**) describes OPENORCA and MOPENORCA.

Fine-tuning Technique. We perform full fine-tuning with bf16 precision to study the effects of full fine-tuning with multilingual data in Phase 2 and its effect on task ability. We also wish to exploit the benefits gained via complete fine-tuning of these models, which may not be possible with parameter efficient fine-tuning [1, 32]. However, in §5, we propose a heuristic-based layer freezing strategy to mitigate forgetting of task ability in which we freeze some layers and fine-tune the rest. For our

Model	Phase 1 Dataset	Phase 2 Dataset	MLQA (↑) Phase 1	MLQA (↑) Phase 2	XLSUM (↑) Phase 1	XLSUM (↑) Phase 2	XQuAD (↑) Phase 1	XQuAD (↑) Phase 2	Average Phase 1	Average Phase 2
MISTRAL-7B	ALPACA	MULTIALPACA	0.229	0.288	0.012	0.060	0.290	0.602	0.177	0.317
	Instruct		0.246	0.307	0.012	0.033	0.351	0.436	0.203	0.259
LLAMA-3-8B	ALPACA		0.438	0.597	0.033	0.034	0.586	0.737	0.352	0.456
	Instruct		0.609	0.321	0.048	0.027	0.712	0.417	0.456	0.255

Table 2: Language Ability results for two-phase Continual Fine-tuning (CFT). With green, we denote an improvement in language ability post Phase 2 fine-tuning. Likewise, we denote a decline in language ability with red. For MLQA and XQUAD we use F1 abstractive score, while for XLSUM we use ROUGE Score.

experiments, we use *Axolotl*[2], an open-source framework to fine-tune LLMs. We conducted our experiments on NVIDIA A100 GPUs with 80 GB RAM.

Evaluation Tasks. To quantify an LLM's task ability, we evaluate Phase 1 and Phase 2 models on two instruction-following tasks (i) `IFEval` [49] and (ii) `Alpaca Eval` [28], (iii) `MMLU` [19] for problem-solving and (iv) `HellaSwag` [45] for commonsense reasoning ability. To quantify an LLM's language ability, we evaluate our fine-tuned models on three benchmark datasets comprising two multilingual generative tasks: question answering (`MLQA` [27] & `XQuAD` [3]) and summarization (`XLSUM` [17]). Further details on these tasks are available in the Appendix (§**??**).

To evaluate our models on TA and LA, we use *LM-Evaluation-Harness*[3], which is a unified framework for zero/few-shot evaluations of LLMs. For both task and language ability, we use **zero-shot** evaluation. For additional details on the training setup, code, and evaluation tasks, we refer the reader to the Appendix (§A).

4.2 Task and Language Ability Results

We compare the task and language ability of MISTRAL-7B and LLAMA-3-8B continually fine-tuned models on different phase-wise datasets[4]. Table 1 presents the results for task ability, while Table 2 presents the results for language ability. Table 2 reports the average score across languages. We provide language-specific scores in the Appendix (§B).

Results Discussion. From Table 1, we see that for phase-wise datasets like Instruct and MULTIALPACA, the performance of the Phase 2 models trained on them declines for English. This decline occurs when they are continually fine-tuned on multilingual data in Phase 2. However, we see a jump in MISTRAL-7B's language ability from the results for the multilingual generative tasks (Table 2). These models fine-tuned on multilingual datasets show catastrophic forgetting in English. However, for phase-wise datasets like ALPACA followed by MULTIALPACA, we see that models trained on them do not show a decline in task ability (Table 1). We also see a gain in these models' language ability (Table 2)[5].

Additional Ablations. In the Appendix (§B), we also present results for OPENORCA-MOPENORCA phase-wise datasets. For MISTRAL-7B, we observe that the average task ability of the Phase 2 model (over Phase 1's MISTRAL-7B-OPENORCA) marginally declines: 0.487 from 0.504. Whereas, for MISTRAL-7B-INSTRUCT, the average decline in task ability is significant: 0.376 from 0.529. Likewise, for LLAMA-3-8B, the average task ability for LLAMA-3-8B OPENORCA MOPENORCA sees an increase of 0.415 from 0.404. In contrast, with Instruct-MOPENORCA as the phase-wise datasets, the task ability significantly drops, from 0.437 to 0.173.

Observation. With Table 1, we see that our two-phase CFT setup for language adaption shows an interesting trend: for certain pairs of phase-wise datasets (e.g., ALPACA & MULTIALPACA), the LLM after Phase 2 sees an improvement in the task ability (computed on English evaluation tasks). We notice that phase-wise datasets like ALPACA and MULTIALPACA have the same seed prompts. Alternately, the two datasets encode the same tasks in different languages. We hypothesize an LLM

[2] `github.com/axolotl-ai-cloud/axolotl/`

[3] `github.com/EleutherAI/lm-evaluation-harness`

[4] When it is clear from the context, we use "Instruct" to denote the dataset used in Phase 1 to instruction fine-tune MISTRAL-7B-INSTRUCT or LLAMA-3-8B-INSTRUCT.

[5] LLAMA-3-8B Instruct MULTIALPACA shows deterioration in LA. We explain this behavior in §5.3.

fine-tuned on either of these datasets learns the same task ability, and therefore, the second phase of CFT leads to lesser interference in the representation space. That is, an LLM continually fine-tuned on ALPACA & MULTIALPACA preserves its task ability across phases. We next define two metrics that aim to quantify the task-specific similarity of two datasets.

4.3 Phase-wise Datasets: Similarity of Representations

Dataset Embedding Similarity (DES). To quantify whether two datasets encode the same tasks, we define DES that computes a similarity score using the dot product of the average representations (embedding) generated by a language-agnostic model.

Definition 1 (Dataset Embedding Similarity (DES))**.** *Given a language-agnostic text embedding model* Θ*, and any pair of datasets* D_1 *and* D_2*, let DES be the function* $f_{DES} : D \times D \to [0, 1]$

$$f_{DES}(D_1, D_2; \Theta) = \langle \mathbf{E}_\Theta(D_1), \mathbf{E}_\Theta(D_2) \rangle \tag{1}$$

Here, $\mathbf{E}_\Theta(D_i) \in \mathbb{R}^d,\ \forall i \in \{1, 2\}$ *is the normalized mean embedding across samples in* D_i*.*

Higher the DES score, more similar the embedding, i.e., greater similarity between D_1 and D_2. For Θ, we use the language-agnostic sentence-tokenizer LaBSE [13]. We compute DES by encoding 500 random samples from ALPACA, MULTIALPACA, OPENORCA, and MOPENORCA, and measure f_{DES} for each pair.

Fixing ALPACA as the Phase 1 dataset D_1, when the Phase 2 dataset D_2 is MULTIALPACA, the DES score is 0.924 and 0.792 for MOPENORCA. When D_1 is OPENORCA, the DES score for MOPENORCA as D_2 is 0.953 and 0.774 when D_2 is MULTIALPACA. For dataset pairs with similar tasks, we see a high DES score and relatively low scores for datasets with different tasks. That is, DES captures the (pair-wise) variation in task abilities of these datasets.

Model Parameter Difference (MPD). Another method of quantifying the similarity of tasks for two datasets D_1 and D_2 is to compute the difference between the parameters of models Θ_1 (fine-tuned on D_1) and Θ_2 (fine-tuned on D_2). Geometrically, the difference of the parameters captures the representation shift of Θ_2 in the space defined by Θ_1. If D_1 & D_2 encode the same tasks, the combined shift by Θ_2 should be relatively lower, compared to the shift if D_1 & D_2 encode different tasks. Formally,

Definition 2 (Model Parameter Difference (MPD))**.** *Given any two models* Θ_1 *and* Θ_2 *fine-tuned on self-instruct datasets* D_1 *and* D_2 *respectively, from the same base model* Θ_B*, let MPD be the function* $f_{MPD} : \Theta \times \Theta \to \mathbb{R}_{\geq 0}$ *s.t.*

$$f_{MPD}(\Theta_1, \Theta_2; \Theta_B) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{w}(\Theta_{1,i}) - \mathbf{w}(\Theta_{2,i})\|_2 \tag{2}$$

Here, $\mathbf{w}(\Theta_{j,i}),\ \forall j \in \{1, 2\}$ *is* Θ_j*'s* i^{th} *parameter.*

The smaller the MPD score, the closer the fine-tuned models are in the parameter space. Fixing MISTRAL-7B as the base model Θ_B, and D_1 as MULTIALPACA, we vary D_2 as one of ALPACA, OPENORCA, and MOPENORCA, and observe the corresponding MPD scores. We normalize the MPD scores with the maximum observed score across all three models for a fair comparison. With D_2 as ALPACA, the MPD score is 0.294. For D_2 as Instruct, MPD is 1.0 and 0.55 when D_2 is OPENORCA. These scores show a similar trend to DES: for ALPACAand MULTIALPACA, the scores are lower, highlighting the similarities in the datasets in the parameter space. We see relatively higher scores for the other pair of models, implying a difference in the dataset pairs.

4.4 Visualizing Decline in Task Ability

Setup. To explain the effect of similar phase-wise data sets on the LLM task ability, we look at the model representations when parsing English (as the task ability is computed over English). We feed MTBENCH [48] to the models, an English prompt dataset for testing, and visualize the similarity between the mean hidden activations, for each model layer. For the analysis, given an LLM Θ with l layers, let $X_\Theta \in \mathbb{R}^{l \times d}$ be the mean hidden activations, across n samples from MTBENCH.

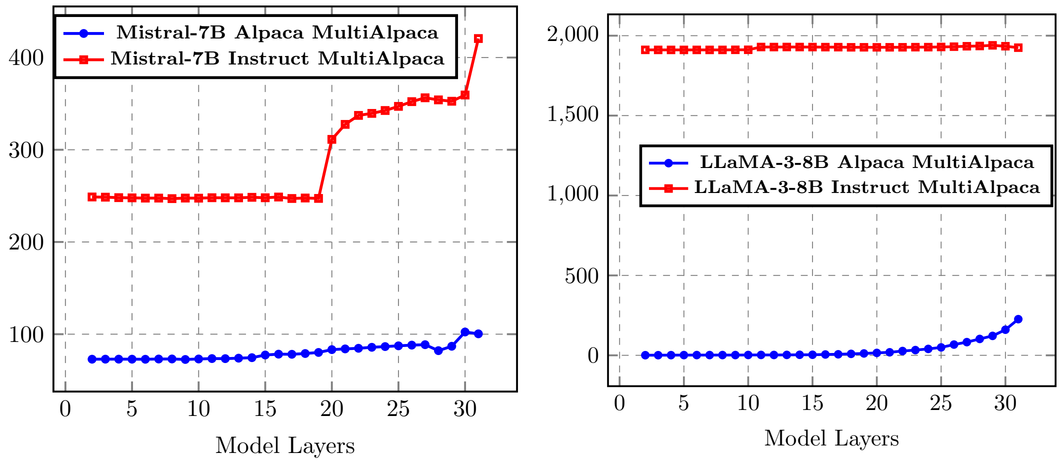

Figure 2: We see a greater change in the variation of the representations for non-similar datasets (e.g., Instruct & MULTIALPACA) compared to similar datasets (e.g., ALPACA & MULTIALPACA). Interestingly, for LLAMA-3-8B the change is large across layers and a magnitude higher than MISTRAL-7B. For MISTRAL-7B, we see the later layers showing the most change.

t-SNE Visualization. Figure 1 depicts t-SNEs [42] for $X_{\text{MISTRAL-7B}}$ and $X_{\text{LLAMA-3-8B}}$ LLMs, continually fine-tuned on the phase-wise datasets ALPACA & MULTIALPACA and Instruct & MULTIALPACA. We observe that for similar phase-wise datasets, the model before and after Phase 2 produces similar hidden activations. Contrarily, for non-similar phase-wise datasets, the hidden activations form distinct clusters, implying separation between the phase-wise activations. That is, the model representations for non-similar phase-wise datasets are well-separated. The model representations during Phase 2 do not align with Phase 1 representations; thus, resulting in greater model weight interference leading to a decline in task ability.

Visualizing Variance in Model Representations. Figure 1 provides some intuition for the correlation between phase-wise datasets and decline in task ability. To further understand the layer-wise behavior of the hidden activations, similar to Chang et al. [8], we compute covariance matrices Σ_Θ for each X_Θ. Intuitively, Σ_Θ captures the variance in different directions for representations of hidden activations for Θ.

We first compute the mean centered activation matrix $\bar{X}_\Theta = X_\Theta - \mu_\Theta$, according to $\mu_\Theta \in \mathbb{R}^d$. Next, we derive $\Sigma_\Theta = \frac{1}{l-1} \cdot \bar{X}_\Theta^T \bar{X}_\Theta \in \mathbb{R}^{d \times d}$. To compare the layer-wise variance in representations, we compute the L2-Norm of the difference of the matrices $\Sigma_{\text{MISTRAL-7B}}$ (Figure 2 **(left)**) or $\Sigma_{\text{LLAMA-3-8B}}$ (Figure 2 **(right)**) when continually fine-tuned on ALPACA & MULTIALPACA (blue lines) or Instruct & MULTIALPACA (red lines).

From the figures, we see clear evidence of representational change, both in terms of the magnitude of the change and the subset of layers that show a greater change. For MISTRAL-7B, the Phase 2 model after CFT with Instruct & MULTIALPACA, shows 3 to 4 times more variation in its representations compared to the model with ALPACA & MULTIALPACA phase-wise datasets. This gap is significantly larger for LLAMA-3-8B.

5 Mitigating Strategies for Multilingual CFT

To mitigate the decline in task ability, we study two CFT techniques, Generative Replay (`GR`) and heuristic-based Layer Freezing (`LF`). In Generative Replay, we consider a new English data generation method motivated by the correlation between dataset similarity and task ability (§4.2). With heuristic-based Layer Freezing, we employ specific heuristics to find out the subset of layers to freeze in the model during Phase 2 fine-tuning.

CFT Setup			Task Ability (TA)					Language Ability (LA)			
	Phase 2 Dataset	Mitigating Strategy	IFEval (↑)	Alpaca Eval (↑)	MMLU (↑)	HellaSwag (↑)	Avg (↑)	MLQA (↑)	XLSum (↑)	XQUAD (↑)	Avg (↑)
MISTRAL-7B	–	–	0.55	**0.35**	0.575	**0.641**	0.529	0.246	0.012	0.351	0.203
	MULTIALPACA	–	0.462	0.15	0.533	0.416	0.390	0.307	0.033	0.436	0.259
		LF_H1	0.456	0.03	0.497	0.598	0.395	0.176	0.016	0.215	0.136
		LF_H2	0.364	0.12	0.364	0.504	0.338	0.213	0.014	0.442	0.223
		GR_5	0.540	0.17	0.540	0.611	0.465	**0.311**	0.008	0.428	0.249
		GR_10	0.567	0.12	0.567	0.594	0.462	0.213	0.007	0.427	0.215
		LoRA	0.383	0.09	0.579	0.625	0.42	0.289	**0.043**	**0.518**	**0.283**
		ER_10	**0.593**	0.08	**0.580**	0.635	**0.599**	0.249	0.008	0.398	0.218
LLAMA-3-8B	–	–	**0.735**	**0.14**	0.340	0.533	**0.436**	**0.609**	**0.048**	**0.712**	**0.456**
	MULTIALPACA	–	0.182	0.10	0.239	0.278	0.217	0.321	0.030	0.417	0.256
		LF_H1	0.303	0.0	0.231	0.275	0.202	0.368	0.037	0.505	0.303
		LF_H2	0.380	0.06	0.485	0.525	0.373	0.400	0.038	0.505	0.314
		GR_5	0.269	0.01	0.516	0.316	0.279	0.437	0.019	0.593	0.349
		GR_10	0.264	0.12	0.229	0.250	0.228	0.254	0.009	0.314	0.192
		LoRA	0.196	0.0	0.280	0.235	0.179	0.007	0.008	0.005	0.007
		ER_10	0.420	0.02	**0.603**	**0.561**	0.420	0.434	0.025	0.53	0.330

Table 3: Task and Language ability results for our mitigating strategies, Generative Replay (GR_5 & GR_10) and Layer Freezing (LF_H1 & LF_H2). We also use LoRA [20] and ER_10 as two baseline strategies. Here, we perform Phase 2 fine-tuning with rank 64 and quantisation bfloat16 for LoRA. For ER_10, we use the English dataset used in GR_5 with original responses. *The Phase 1 dataset is Instruct for each row.* The first two rows for both MISTRAL-7B and LLAMA-3-8B provide numbers for Instruct and Instruct-MULTIALPACA (from Table 1 & Table 2).

5.1 Generative Replay

Typically, Generative Replay (GR) is a technique that generates data from past distributions to be used alongside new task data for the continual fine-tuning of a model on a new task [38]. However, from §4.2, we see that if the phase-wise datasets encode similar tasks, the decline in task ability is mitigated. Based on this observation, we use the Phase 1 model to generate responses, in English, from the English counterpart of the multilingual dataset used for training in Phase 2. This generated replay dataset acts as a bridge between the distributions of Phase 1 and Phase 2.

During Phase 2 fine-tuning, we include varying quantities of this generated data: specifically, 5% (GR_5) and 10% (GR_10), of the Phase 2 dataset. As a **baseline**, we also fine-tune the models with a similar sized subset of the English counterpart with original responses[6]. We refer to this baseline as English Replay (ER_10).

5.2 Heuristic-based Layer Freezing

Model regularization is an effective technique to mitigate the drop in the previous task's performance in continual learning (e.g., EWC [26]). However, this is computationally inefficient as it requires using both the old and new sets of parameters. Instead, we use Layer Freezing (LF), a relatively efficient technique for use as a 'regularizer' to preserve task ability during Phase 2. We consider the following two variations to select the set of layers to freeze:

1. LF_H1: freezing a random set of 10 layers of the model from Phase 1 to be fine-tuned in Phase 2.
2. LF_H2: freezing the top-10 layers that have changed the most during Phase 1 fine-tuning (e.g., MISTRAL-7B Base to MISTRAL-7B-INSTRUCT). We select these layers separately for Key, Query, and Value, for each attention head.

We present our results in Table 3 for both GR and LF. Along with English Replay (ER), we define another **baseline** in which we use LoRA [20] for continually fine-tuning in Phase 2.

[6]This dataset may not be available for all multilingual datasets eg. Aya [39]. In that case, instructions can always be translated to English but it is not always practical to translate responses. Hence, this baseline is the best-case scenario for our GR strategy.

5.3 Results Discussion

From Table 3, we see that GR and LF successfully mitigate the decline in task ability and also show gains in language ability. For instance, MISTRAL-7B with GR_5 achieves better performance in MLQA and XLSUM when fine-tuned with MULTIALPACA. We also close the gap with MISTRAL-7B-INSTRUCT on IFEval, Alpaca Eval, MMLU, and HellaSwag with our mitigation strategies.

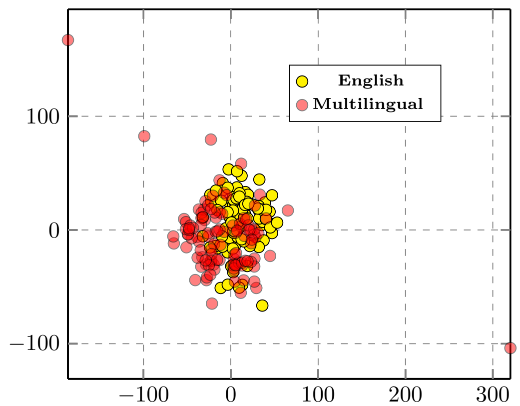

Figure 3: *
(a) MISTRAL-7B

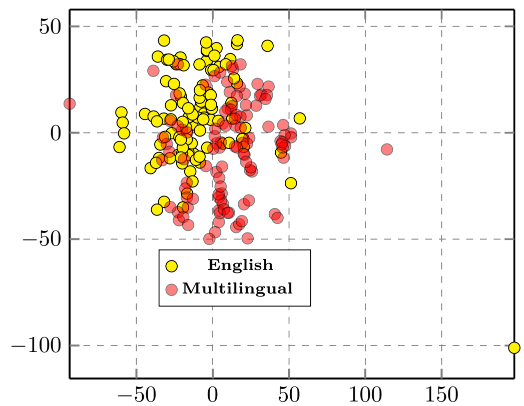

Figure 4: *
(b) LLAMA-3-8B

Figure 5: Demonstrating extent of cross-lingual transfer in MISTRAL-7B and LLAMA-3-8B on a parallel dataset prepared by subsampling FLORES [10]. We find that the English activation cluster for LLAMA-3-8B is separated from the multilingual cluster, compared to MISTRAL-7B.

LLAMA-3-8B Doesn't Show Consistent Improvement with our Mitigations. From Table 3, while both GR and LF improve on the baseline LLAMA-3-8B-INSTRUCT MULTIALPACA, the gains in task and language ability are not comparable to LLAMA-3-8B-INSTRUCT.

To understand this further, for GR, we investigate the cross-linguality difference between LLAMA-3-8B and MISTRAL-7B. Like Figure 1, we plot t-SNEs of the mean model activations for the MISTRAL-7B and LLAMA-3-8B base models on two parallel datasets, English and Multilingual. We create the parallel datasets by subsampling data from FLORES [10]. In Figure 5, we see that the English activation cluster for LLAMA-3-8B is separated out from multilingual cluster, compared to MISTRAL-7B. This suggests that GR may not be as effective when the model has less cross lingual ability. While for LF, we acknowledge that our method to identify the layers to freeze may not be the best and better methods to identify which layers to freeze can be a direction for future work.

Last, but not the least, we acknowledge that LLAMA-3-8B-INSTRUCT seems to be a strong model even on multilingual benchmarks. Hence, it is also important to evaluate Phase 1 models on these benchmarks first and then decide if the Phase 2 fine-tuning step should be undertaken or not.

With regards to LLAMA-3-8B-INSTRUCT MULTIALPACA LA results in Table 2, we believe that this is due to lack of cross-linguality in LLAMA-3-8B-INSTRUCT and less data in MULTIALPACA which fails to cause sufficient representation drift to improve the model's performance.

Forgetting with LoRA. For MISTRAL-7B-INSTRUCT and LoRA fine-tuning, we see an increase in language ability but a decline in task ability. But the decline is not as much as full fine-tuning. For LLAMA-3-8B-INSTRUCT and LoRA, there is a greater decline in both task and language ability. The decline is similar (or slightly lower) than the full fine-tuning scenario. These results show that LoRA also suffers from forgetting when used for continual fine-tuning.

Additional Results. In the Appendix (§C), we repeat the same experiment from §4.4 to quantify the representation change in the fine-tuned models using our mitigating strategies. We see a trend similar to Figure 2. That is, a decrease in the variation in the model activations, compared to the baseline model trained on Instruct and MULTIALPACA. The decrease is more pronounced for MISTRAL-7B compared to LLAMA-3-8B. In Appendix §C, we also present TA and LA results for the Instruct-MOPENORCA phase-wise datasets.

6 Conclusion & Future Work

In this paper, to the best of our knowledge, we present a first study on the influence of the similarity of phase-wise datasets on the task and language adaptability of LLMs through CFT. Through extensive experiments on the MISTRAL-7B and LLAMA-3-8B models, we show that when datasets are similar, task ability is preserved; otherwise, it declines. Towards mitigation, we study layer freezing and generative replay as mitigating strategies based on specific heuristics. Our results indicate that these strategies help improve task performance while not compromising on the LLM's language adaptability.

Future Work. Our results show that there is no one-size-fits-all strategy to mitigate decline in task ability, among the strategies discussed. Future work can explore developing other parameter-efficient regularization methods that address the current computational challenges with methods like EWC or forgetting due to LoRA. One can also explore analytical notions for task similarity in datasets.

7 Limitations

The study assumes that the similarity between phase-wise datasets can be effectively quantified using DES and MPD metrics. However, these metrics may not capture all nuances of task similarity. Moreover, the experiments were conducted on MISTRAL-7B and LLAMA-3-8B models. The results and conclusions drawn may not generalize to other LLMs with different architectures or training paradigms. Additionally, The study's fine-tuning and evaluation processes were constrained by available computational resources. More extensive experiments with larger models and longer training datasets were not possible.Furthermore, while generative replay and heuristic-based layer freezing showed promise, their effectiveness may vary with different models and datasets. Lastly, the evaluation of task and language ability was based on specific benchmarks. These metrics may not encompass all aspects of model performance, particularly in real-world applications.

References

[1] Divyanshu Aggarwal, Ashutosh Sathe, and Sunayana Sitaram. Maple: Multilingual evaluation of parameter efficient finetuning of large language models. *arXiv preprint arXiv:2401.07598*, 2024.

[2] Spurthi Amba Hombaiah, Tao Chen, Mingyang Zhang, Michael Bendersky, and Marc Najork. Dynamic language models for continuously evolving content. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2514–2524, 2021.

[3] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856, 2019.

[4] Kartikeya Badola, Shachi Dave, and Partha Talukdar. Parameter-efficient finetuning for robust continual multilingual learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9763–9780, July 2023.

[5] Terra Blevins, Tomasz Limisiewicz, Suchin Gururangan, Margaret Li, Hila Gonen, Noah A Smith, and Luke Zettlemoyer. Breaking the curse of multilinguality with cross-lingual expert language models. *arXiv preprint arXiv:2401.10440*, 2024.

[6] Samuel Cahyawijaya, Holy Lovenia, Tiezheng Yu, Willy Chung, and Pascale Fung. Instructalign: High-and-low resource language alignment via continual crosslingual instruction tuning. In *Proceedings of the First Workshop in South East Asian Language Processing*, pages 55–78, 2023.

[7] Salvador Carrión and Francisco Casacuberta. Few-shot regularization to tackle catastrophic forgetting in multilingual machine translation. In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 188–199, 2022.

[8] Tyler Chang, Zhuowen Tu, and Benjamin Bergen. The geometry of multilingual language model representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing EMNLP*, pages 119–136, 2022.

[9] Yihong Chen, Kelly Marchisio, Roberta Raileanu, David Adelani, Pontus Lars Erik Saito Stenetorp, Sebastian Riedel, and Mikel Artetxe. Improving language plasticity via pretraining with active forgetting. *Advances in Neural Information Processing Systems*, 36:31543–31557, 2023.

[10] Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

[11] Tejas Dhamecha, Rudra Murthy, Samarth Bharadwaj, Karthik Sankaranarayanan, and Pushpak Bhattacharyya. Role of language relatedness in multilingual fine-tuning of language models: A case study in indo-aryan languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8584–8595, 2021.

[12] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo

Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun

Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

[13] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding, 2020.

[14] Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities. *arXiv preprint arXiv:2404.17790*, 2024.

[15] Zheng Gong, Kun Zhou, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. Continual pre-training of language models for math problem understanding with syntax-aware memory network. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics ACL*, pages 5923–5933, 2022.

[16] Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to (re) warm your model? *arXiv preprint arXiv:2308.04014*, 2023.

[17] Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online, August 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.findings-acl.413`.

[18] Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models. *arXiv preprint arXiv:2311.16206*, 2023.

[19] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

[21] Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. *arXiv preprint arXiv:2204.14211*, 2022.

[22] Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, KIM Gyeonghun, Stanley Jungkyu Choi, and Minjoon Seo. Towards continual knowledge learning of language models. In *International Conference on Learning Representations ICLR*, 2022.

[23] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

[24] Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4764–4780, 2022.

[25] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

[26] 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.

[27] Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*, 2019.

[28] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

[29] Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/Open-Orca/OpenOrca`, 2023.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[31] Jishnu Mukhoti, Yarin Gal, Philip HS Torr, and Puneet K Dokania. Fine-tuning can cripple your foundation model; preserving features may be the solution. *arXiv preprint arXiv:2308.13320*, 2023.

[32] Ashwinee Panda, Berivan Isik, Xiangyu Qi, Sanmi Koyejo, Tsachy Weissman, and Prateek Mittal. Lottery ticket adaptation: Mitigating destructive interference in llms, 2024. URL `https://arxiv.org/abs/2406.16797`.

[33] Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Mad-x: An adapter-based framework for multi-task cross-lingual transfer, 2020. URL `https://arxiv.org/abs/2005.00052`.

[34] Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, 2022.

[35] Karan Praharaj and Irina Matveeva. Multilingual continual learning approaches for text classification. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 864–870, 2023.

[36] Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. Multilingual instruction tuning with just a pinch of multilinguality. *arXiv preprint arXiv:2401.01854*, 2024.

[37] Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, and Hao Wang. Continual learning of large language models: A comprehensive survey. *arXiv preprint arXiv:2404.16789*, 2024.

[38] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay, 2017. URL `https://arxiv.org/abs/1705.08690`.

[39] Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Vu Minh Chien, Sebastian Ruder, Surya Guthikonda, Emad A. Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. Aya dataset: An open-access collection for multilingual instruction tuning, 2024. URL `https://arxiv.org/abs/2402.06619`.

[40] Alane Suhr and Yoav Artzi. Continual learning for instruction following from realtime feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 32340–32359, 2023.

[41] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[42] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[43] Xiangpeng Wei, Haoran Wei, Huan Lin, Tianhao Li, Pei Zhang, Xingzhang Ren, Mei Li, Yu Wan, Zhiwei Cao, Binbin Xie, Tianxiang Hu, Shangjie Li, Binyuan Hui, Bowen Yu, Dayiheng Liu, Baosong Yang, Fei Huang, and Jun Xie. Polylm: An open source polyglot large language model, 2023. URL `https://arxiv.org/abs/2307.06018`.

[44] Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. Efficient continual pre-training for building domain specific large language models. *arXiv preprint arXiv:2311.08545*, 2023.

[45] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[46] Han Zhang, Lin Gui, Yuanzhao Zhai, Hui Wang, Yu Lei, and Ruifeng Xu. Copf: Continual learning human preference through optimal policy fitting. *arXiv preprint arXiv:2310.15694*, 2023.

[47] Yiran Zhao, Wenxuan Zhang, Huiming Wang, Kenji Kawaguchi, and Lidong Bing. Adamergex: Cross-lingual transfer with large language models via adaptive adapter merging, 2024. URL `https://arxiv.org/abs/2402.18913`.

[48] 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. *Advances in Neural Information Processing Systems*, 36, 2024.

[49] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Training Details

A.1 Hyperparameters for Fine-tuning and Training Setup

Hyperparameter	Value
Learning Rate	1×10^{-6}
Epochs	4
Global Batch size	16
Scheduler	Cosine
Warmup	Linear
Warmup Steps	10
Optiimizer	AdamW [30]
Weight Decay	0

Table 4: Hyperparameters for continual fine-tuning

A.2 Fine-tuning and Evaluation Dataset Details

A.3 Evaluation Tasks

In this paper, we consider two sets of benchmarks to evaluate task and language ability. We explain them briefly next.

Task Ability (TA). To quantify an LLM's task ability, we evaluate Phase 1 and Phase 2 models on the following tasks:

1. `IFEval` [49]: Instruction-Following Evaluation (`IFEval`) asses the ability of an LLM to follow natural language instructions. It comprises 500 verifiable instructions (e.g., "*mention the keyword AI 3 times*"). We choose `IFEval` as the instructions are verifiable and also test an LLM's context understanding.
2. `Alpaca Eval` [28]: This is an LLM based automatic evaluator for instruction following models, to measure task ability. Like Aggarwal et al. [1], we evaluate our CFT models against *text-davinci-003* responses on 800 instructions and use GPT4 (*gpt-4-32k)* as the evaluator.
3. `MMLU` [19]: Massive Multitask Language Understanding (`MMLU`) is a benchmark to assess an LLM's knowledge and problem solving abilities. It includes 57 subjects across domains like STEM, or law, with 16k MCQs in total.
4. `HellaSwag` [45]: This is a popular benchmark to evaluate the commonsense reasoning ability of an LLM. `HellaSwag`'s test split contains 10k samples in total.

Language Ability (LA). To quantify an LLM's language ability, we evaluate our fine-tuned models on three benchmark datasets comprising two multilingual generative tasks: question answering and summarisation.

- **Question Answering:** `MLQA` [27] contains 5k extractive question-answering instances in 7 languages. The `XQuAD` dataset [3] consists of a subset of 240 paragraphs and 1190 question-answer pairs across 11 languages.
- **Summarisation:** `XLSUM` [17] spans 45 languages, and we evaluate our models in Arabic, Chinese-Simplified, English, French, Hindi, Japanese and Spanish.

To evaluate our models on TA and LA, we use *LM-Evaluation-Harness*[7], which is a unified framework for zero/few-shot evaluations of LLMs. For both task and language ability, we use **zero-shot** evaluation. For additional details on the training setup, code and the evaluation tasks, we refer the reader to the accompanying supplementary (Appendix §A). A reproducibility checklist is available after the References section with details in Appendix §A.

[7] `https://github.com/EleutherAI/lm-evaluation-harness`

Model	Phase 1 (P1) Dataset	Phase 2 (P2) Dataset	IFEval (↑)		Alpaca Eval (↑)		MMLU (↑)		HellaSwag (↑)		Average	
			P1	P2	P1	P2	P1	P2	P1	P2	P1	P2
MISTRAL-7B	OPENORCA	MOPENORCA	0.494	0.482	0.31	0.32	0.601	0.582	0.612	0.562	0.504	0.487
	Instruct		0.550	0.426	0.35	0.06	0.575	0.507	0.641	0.509	0.529	0.376
LLAMA-3-8B	OPENORCA		0.377	0.425	0.09	0.07	0.579	0.599	0.571	0.564	0.404	0.415
	Instruct		0.735	0.205	0.14	0.0	0.340	0.236	0.533	0.250	0.437	0.173

Table B1: Task Ability results for two-phase Continual Fine-tuning (CFT). With green, we highlight an increase in a model's task ability post P2 fine-tuning. Likewise, red highlights a decline in a model's task ability.

Model	Phase 1 Dataset	Phase 2 Dataset	MLQA Phase 1						MLQA Phase 2					
			ar	de	es	hi	vi	zh	ar	de	es	hi	vi	zh
MISTRAL-7B	ALPACA	MULTIALPACA	0.143	0.337	0.331	0.149	0.385	0.031	0.172	0.485	0.529	0.196	0.336	0.009
	Instruct		0.113	0.440	0.395	0.088	0.369	0.073	0.228	0.456	0.529	0.279	0.327	0.0222
LLAMA-3-8B	ALPACA		0.320	0.538	0.563	0.438	0.611	0.155	0.552	0.672	0.765	0.573	0.784	0.237
	Instruct		0.549	0.701	0.769	0.624	0.788	0.192	0.316	0.453	0.526	0.137	0.464	0.028
MISTRAL-7B	OPENORCA	MOPENORCA	0.374	0.504	0.511	0.395	0.600	0.226	0.298	0.506	0.572	0.274	0.481	0.030
	Instruct		0.113	0.440	0.395	0.088	0.369	0.073	0.115	0.253	0.213	0.088	0.222	0.038
LLAMA-3-8B	OPENORCA		0.262	0.545	0.565	0.369	0.568	0.099	0.437	0.549	0.622	0.462	0.625	0.024
	Instruct		0.320	0.538	0.563	0.438	0.611	0.155	0.554	0.701	0.771	0.625	0.787	0.188

Table B2: MLQA: Language Ability results for two-phase Continual Fine-tuning (CFT).

B Evaluating Language Ability for Multilingual Continual Fine-tuning

Task Ability. Table B1 and Table **??** present the task and language ability numbers of our ablations on the OPENORCA and MULTIALPACA datasets using MISTRAL-7B and LLAMA-3-8B models.

Language Ability. Table B2, Table B3, and Table B4 present the language-specific results for MLQA, XLSUM, and XQuAD, respectively.

C Mitigating Strategies

Visualizing Variance in Model Representations. In Figure 6, we repeat the same experiment as in § 4.5 to quantify the representation change in the fine-tuned models using our mitigating strategies. The trend seen is expected from § 4.5: we see a decrease in the variation in the model activations, compared to the baseline model trained on Instruct and MULTIALPACA. The decrease is more pronounced for MISTRAL-7B compared to LLAMA-3-8B.

Additional Ablations. We also present the impact of our mitigating strategies for the Instruct-MOPENORCA phase-wise datasets. Table C5 presents these results.

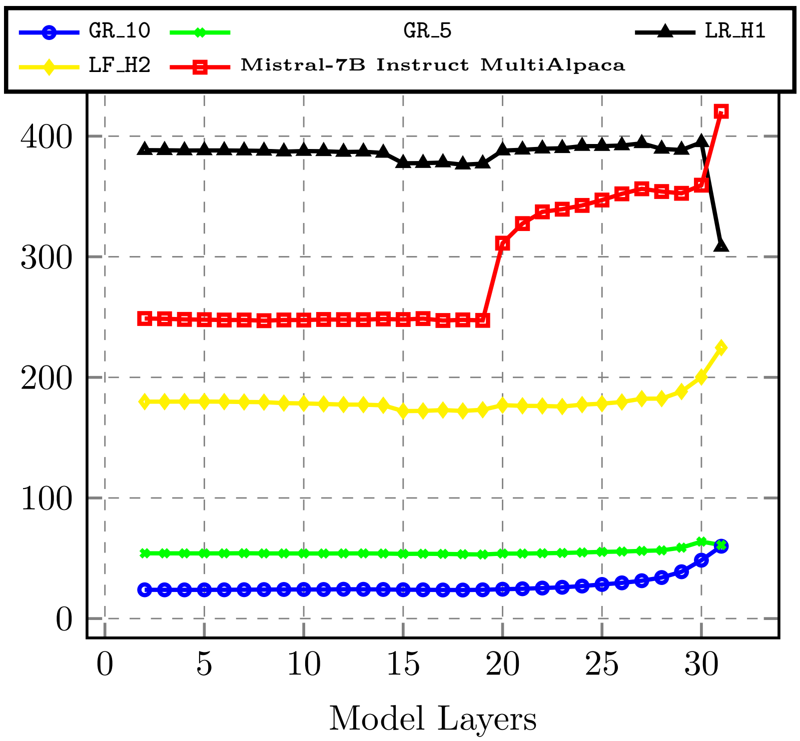

Figure 6: **Visualizing Variance in Model Representations for MISTRAL-7B Mitigating Strategies:** We see a decrease in the variance of model representations for models trained using our mitigation strategies compared to vanilla Phase 2 models (similar to Figure 2).

Model	Phase 1 Dataset	Phase 2 Dataset	XLSUM Phase 1 Arabic	Phase 1 Chinese_simplified	Phase 1 french	Phase 1 Hindi	Phase 1 Japanese	Phase 1 Spanish	Phase 2 Arabic	Phase 2 Chinese_simplified	Phase 2 french	Phase 2 Hindi	Phase 2 Japanese	Phase 2 Spanish
MISTRAL-7B	ALPACA	MULTIALPACA	0.001	0.012	0.025	0.001	0.012	0.023	0.022	0.034	0.112	0.016	0.067	0.106
	Instruct		0.001	0.005	0.028	0.001	0.009	0.025	0.016	0.015	0.060	0.010	0.040	0.056
LLAMA-3-8B	ALPACA		0.005	0.015	0.071	0.003	0.037	0.067	0.003	0.018	0.073	0.002	0.041	0.070
	Instruct		0.008	0.015	0.092	0.004	0.080	0.087	0.002	0.013	0.055	0.001	0.055	0.051
MISTRAL-7B	OPENORCA	MOPENORCA	0.001	0.010	0.014	0.001	0.007	0.009	0.001	0.006	0.018	0.001	0.008	0.016
	Instruct		0.001	0.005	0.028	0.001	0.009	0.025	0.007	0.017	0.092	0.005	0.030	0.088
LLAMA-3-8B	OPENORCA		0.000	0.003	0.061	0.000	0.004	0.035	0.000	0.003	0.016	0.001	0.000	0.013
	Instruct		0.008	0.015	0.092	0.004	0.080	0.087	0.007	0.015	0.091	0.004	0.082	0.087

Table B3: XLSUM: Language Ability results for two-phase Continual Fine-tuning (CFT).

Model	Phase 1 Dataset	Phase 2 Dataset	XQuAD Phase 1 ar	de	el	es	hi	ro	ru	th	tr	vi	zh	Phase 2 ar	de	el	es	hi	ro	ru	th	tr	vi	zh
Mistral-7B	Alpaca Instruct	MultiAlpaca	0.194	0.379	0.248	0.374	0.224	0.418	0.150	0.185	0.454	0.475	0.088	0.613	0.692	0.657	0.713	0.670	0.679	0.661	0.385	0.666	0.734	0.148
			0.166	0.568	0.260	0.510	0.173	0.508	0.336	0.210	0.460	0.502	0.168	0.369	0.612	0.253	0.634	0.450	0.553	0.555	0.180	0.532	0.566	0.089
LLaMA-3-8B	Alpaca Instruct		0.393	0.689	0.529	0.735	0.644	0.723	0.538	0.398	0.671	0.748	0.376	0.676	0.850	0.710	0.893	0.740	0.817	0.726	0.526	0.770	0.884	0.519
			0.659	0.795	0.702	0.852	0.715	0.810	0.609	0.594	0.728	0.834	0.533	0.444	0.580	0.244	0.657	0.241	0.586	0.493	0.092	0.580	0.558	0.113
Mistral-7B	OpenOrca Instruct	mOpenOrca	0.001	0.010	0.014	0.001	0.007	0.009	0.001	0.006	0.018	0.001	0.008	0.639	0.832	0.570	0.847	0.601	0.776	0.771	0.366	0.734	0.820	0.113
			0.166	0.568	0.260	0.510	0.173	0.508	0.336	0.210	0.460	0.502	0.168	0.256	0.457	0.320	0.443	0.256	0.409	0.215	0.245	0.364	0.428	0.162
LLaMA-3-8B	OpenOrca Instruct		0.505	0.642	0.587	0.711	0.604	0.634	0.651	0.290	0.699	0.685	0.104	0.639	0.832	0.570	0.847	0.601	0.776	0.771	0.366	0.734	0.820	0.113
			0.659	0.795	0.702	0.852	0.715	0.810	0.609	0.594	0.728	0.834	0.533	0.654	0.793	0.703	0.852	0.718	0.808	0.606	0.600	0.729	0.836	0.540

Table B4: XQuAD: Language Ability results for two-phase Continual Fine-tuning (CFT).

CFT Setup				Task Ability					Language Ability			
Model	Phase 1 Dataset	Phase 2 Dataset	Mitigating Strategy	IFEval	ALPACA Eval	MMLU	HellaSwag	Avg	MLQA	XLSum	XQUAD	Avg
MISTRAL-7B	Instruct	–	–	0.55	0.35	0.575	0.641	0.529	0.246	0.012	0.351	0.203
		MOPENORCA	–	0.426	0.06	0.507	0.509	0.376	0.155	0.040	0.323	0.173
			LF_H2	0.401	0.048	0.518	0.487	0.364	0.258	0.060	0.527	0.282
			GR_5	0.281	0.027	0.478	0.495	0.320	0.167	0.042	0.305	0.171
			GR_10	0.305	0.013	0.483	0.494	0.324	0.150	0.038	0.238	0.142
			LoRA	0.587	0.13	0.567	0.591	0.469	0.167	0.027	0.354	0.183
			ER_10	0.367	0.025	0.479	0.493	0.341	0.157	0.042	0.305	0.168

Table C5: Task and Language ability results for our mitigating strategies, Generative Replay (GR_5 & GR_10) and Layer Freezing (LF_H1 & LF_H2). We also use LoRA [20] and ER_10 as two baseline strategies. Here, we perform Phase 2 with rank 64 and bf16 for LoRA. For ER_10, we use the English dataset used in GR_5 with original responses.