Exploring Two-Phase Continual Instruction Fine-tuning for Multilingual Adaptation in Large Language Models

Divyanshu Aggarwal[*1], **Sankarshan Damle**[*1], **Navin Goyal**[2], **Satya Lokam**[1], **Sunayana Sitaram**[2]
[1]Microsoft [2]Microsoft Research India
{divyanshu.aggarwal,t-sandamle,navingo,satya.lokam,sunayana.sitaram}@microsoft.com

Abstract

A key challenge for Large Language Models (LLMs) is improving their Multilingual instruction-following ability over time without deteriorating their ability in languages they already excel at, typically English. In this paper, we study a two-phase *Continual Fine-tuning (CFT)* setup toward improving a model's Multilingual adaptability. Concretely, we consider a two-phase CFT process in which an English-only end-to-end instruction fine-tuned LLM (Phase 1) is sequentially fine-tuned on a multilingual instruction dataset (Phase 2). Across MISTRAL-7B and LLAMA-3-8B and multiple dataset pairs, we show that instructional similarity between phases is critical: aligned datasets preserve or improve English while boosting multilingual ability, whereas misaligned datasets cause English degradation. We show that this degradation arises from representation shift during CFT, and that targeted mitigation strategies, including generative replay and heuristic-based layer freezing, reduce this shift and improve multilingual adaptation.

1 Introduction

The widespread adoption of Large Language Models (LLMs) has led to a growing multilingual user base (Shiyas, 2023). However, ensuring strong performance across languages remains a fundamental challenge, with models consistently performing worse on low-resource languages spoken by millions of speakers worldwide (Ahuja et al., 2023, 2024a). A key limitation is that both labeled and unlabeled training data are predominantly available in English and a few high-resource languages, while resources for other languages, especially low-resource ones, are scarce (Shaham et al., 2024).

Training large models from scratch is computationally expensive, making *fine-tuning* pre-trained LLMs the preferred approach for improving multilingual capabilities (Lankford et al., 2023; Nguyen et al., 2023). A common fine-tuning strategy is to train LLMs on an instruction-following dataset that contains a *mixture* of languages. However, these datasets are often heavily skewed toward English and other high-resource languages, leading to a performance imbalance: models perform strongly in English but struggle with low-resource languages (Dhamecha et al., 2021; Li et al., 2024a,b). Further, prior works show that fine-tuning on a dataset that only contains non-English languages can hurt the model's performance on English due to *catastrophic forgetting* (Wu et al., 2025; Behrouz et al., 2025), which is not desirable for most real-world scenarios due to the volume of English

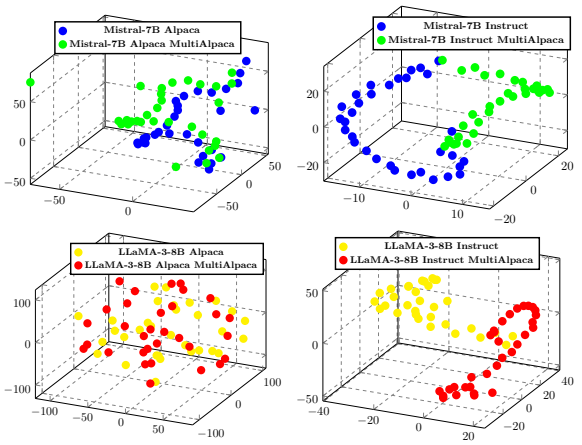


Figure 1: Comparing *t-SNEs* (van der Maaten and Hinton, 2008) of the hidden activations for MISTRAL-7B and LLAMA-3-8B during our two-phase Continual Fine-tuning (CFT) process. We prompt each model with examples from MTBENCH (Zheng et al., 2024), and visualize the similarity between the mean hidden activations, for each model layer. For datasets that encode "similar" instructions (ALPACA & MULTIALPACA), English ability does not decline (e.g., 3% gain for `IFEval`). For non-similar datasets (Instruct & MULTIALPACA), English 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 English ability.

[*]Equal Contribution

Findings of the Association for Computational Linguistics: ACL 2026, pages 31882–31904

queries (Ta, 2023). Ideally, we want the same model to be proficient in both English and other languages to avoid the costs of maintaining multiple models. We refer to an LLM's proficiency in English as its *English Ability* (EA), and its effectiveness across other languages its *Multilingual Ability* (MA). In this work, we aim to improve an LLM's MA while maintaining or improving its EA.

1.1 Our Approach

To bridge the gap between EA and MA, we introduce a *two-phase Continual Fine-tuning* (CFT) setup. We fine-tune a pre-trained LLM on an English instruction dataset in Phase 1 and then fine-tune it on a similarly-sized Multilingual dataset in Phase 2. In Phase 1, we use ALPACA (Taori et al., 2023) and OPENORCA (Lian et al., 2023), and in Phase 2 we use MULTIALPACA (Wei et al., 2023) and MOPENORCA (§4.1). ALPACA and OPENORCA provide high-quality English instruction data, while MULTIALPACA and MOPENORCA are their multilingual counterparts, ensuring consistency in instruction style across phases. To compare the efficacy of our two-phase CFT setup, we compare it with a straightforward single-phase setup where the LLM is fine-tuned on the *mixture* of both the instruction tuning datasets.

We focus on two open-source models, LLAMA-3-8B and MISTRAL-7B as base models for our experiments. We also use fine-tuned versions of them, LLAMA-3-8B-INSTRUCT and MISTRAL-7B-INSTRUCT, as off-the-shelf Phase 1 English fine-tuned models[1]. We quantify a model's English Ability (EA) based on its performance on four English datasets: (i) Two datasets that measure instruction following capabilities (i.e., `IFEval` (Zhou et al., 2023) and `Alpaca Eval` (Li et al., 2023)) and (ii) two that measure reasoning abilities (i.e., MMLU (Hendrycks et al., 2021) and `HellaSwag` (Zellers et al., 2019)). Likewise, we quantify a model's Multilingual Ability (MA) based on its performance on (i) two question-answering tasks (i.e., `MLQA` (Lewis et al., 2019) and `XQuAD` (Artetxe et al., 2019)) and (ii) `XLSUM` (Hasan et al., 2021), a summarization task.

1.2 Our Contributions

CFT Outperforms Mixture. We first observe that models trained using our two-phase CFT setup perform better than the single-phase "dataset mixture" setup (Tables 1, 2; §4.2). Moreover, our two-phase CFT setup overall results in a better model for all languages, including English, for the same number of training steps. The two-phase CFT pipeline also provides more flexibility than training on a mixture of datasets, with the possibility of extending our approach to multi-phase fine-tuning, especially when data from earlier phases might not be available.

Forgetting vs. Dataset Similarity. As mentioned earlier, fine-tuning with multilingual datasets to enhance a model's multilingual ability can lead to a decline in its English ability due to catastrophic forgetting (Mukhoti et al., 2023; Winata et al., 2023). We investigate the factors that may lead to such forgetting by computing the similarity of English and Multilingual Instruction Fine-tuning (IFT) datasets. We observe that when English and multilingual datasets have instructions that are not similar, there is a decline in the Phase 2 model's performance in English. On the other hand, when Phase 1 and Phase 2 datasets encode similar instructions, the Phase 2 model's performance in English improves (refer to Figure 1). 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 English ability (Tables 3, 4; §4.3).

Mitigating Forgetting. We study the efficacy of two tailored variants of existing CFT strategies to mitigate the decline in EA after Phase 2 fine-tuning, while boosting MA. The first strategy is distribution replay. Here, we look at *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. We also try *english replay* which acts as language replay by utilizing existing English parallel data from the Phase 2 distribution. The second strategy employs *layer freezing*. Our heuristic selects specific layers for freezing during Phase 2 fine-tuning based on the weight differences between the Base and Phase 1 models. We also explore Spectrum (Hartford et al., 2024) as an alternative heuristic. We study the gains in EA and MA of these strategies compared to specific baselines (Table 5; §5). To the best of our knowledge, we are the first to explore the effectiveness of CFT on LLMs with multilingual instruction datasets.

[1] LLAMA-3-8B's pre-training data was 5% multilingual, but LLAMA-3-8B-INSTRUCT is primarily non-multilingual (Dubey et al., 2024).

2 Related Work

Continual Learning in LLMs (Chen et al., 2026). 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 (Shi et al., 2024). 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 (Gupta et al., 2023). CPT is employed when distributional shifts occur (i) over time (Amba Hombaiah et al., 2021; Jang et al., 2022a,b), (ii) across languages (Jin et al., 2022; Fujii et al., 2024; Blevins et al., 2024) or (iii) across domains (Ke et al., 2023; Gong et al., 2022; Xie et al., 2023).

On the other hand, CFT involves training the LLM on successive downstream tasks with varying data distribution or time shifts (Shi et al., 2024). CFT comprises fine-tuning for different tasks (Carrión and Casacuberta, 2022; Guan et al., 2025), instruction-tuning (Cahyawijaya et al., 2023; Kang et al., 2025), model refinement/editing (Zhang et al., 2023) and alignment (Suhr and Artzi, 2023). Recent literature also focuses on using CFT to assist the LLM to learn new languages (Praharaj and Matveeva, 2023; Pfeiffer et al., 2022; Badola et al., 2023; Singh et al., 2024c).

CFT: Enhancing LLMs Multilingual Abilities. Cahyawijaya et al. (2023) 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. (2024) introduces multilinguality during the first instruction fine-tuning phase which improves an LLM's instruction following capability across languages. He et al. (2023) 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.

Multilingual Adaptation. 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 (Chen et al., 2023; Zhao et al., 2024; Pfeiffer et al., 2020). For instance, Chen et al. (2023) 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 english ability are either trained in parallel or sequentially. However, in this paper, we try to incorporate multilingual ability in models with the constraint that they may have already learned english 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 Two-phase Continual Fine-tuning Setup

When instruction fine-tuning LLMs, the most natural method is to fine-tune on a "dataset mixture" containing English and Multilingual data (Workshop et al., 2023). However, fine-tuning on all languages simultaneously may introduce performance bias where the model performs better in English (and other high resource languages) (Dhamecha et al., 2021; Li et al., 2024a,b)[2].

Continual Fine-tuning (CFT). To improve the multilingual performance of pre-trained LLMs, we introduce the following two-phase CFT process.

Two-Phase CFT Process

- **Phase 1:** Fine-tune a base LLM end-to-end on an English instruction dataset. Phase 1 aims to teach the LLM *English Instruction Following Ability*, which we refer to as *English Ability* (EA).
- **Phase 2:** Take the fine-tuned LLM from Phase 1 and further fine-tune it end-to-end on a Multilingual instruction dataset. Phase 2 focuses on enhancing the LLM's *Multilingual Ability* (MA), using a dataset with multiple languages and fewer data points per language.

Challenges. The primary challenge in our two-phase CFT process is that the LLM's Multilingual Ability must not come at the cost of its English Ability. We impose *two additional constraints* based on real-world scenarios. First, in Phase 2, we cannot re-use Phase 1's dataset. Often in-

[2] In §4.2, we compare dataset mixture to CFT.

struction fine-tuned LLMs are available without their corresponding datasets (e.g., MISTRAL-7B-INSTRUCT (Jiang et al., 2023)). 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.

4 Evaluating English & Multilingual Ability for Multilingual CFT

4.1 Experiment Setup & Evaluation Tasks

Fine-tuning Models. We continually fine-tune open-source MISTRAL-7B (Jiang et al., 2023) and LLAMA-3-8B (Dubey et al., 2024) LLMs for multilingual adaptation.

Fine-tuning Datasets. For our phase-wise datasets, we use the open-source ALPACA (Taori et al., 2023), MULTIALPACA (Wei et al., 2023), and OPENORCA (Lian et al., 2023) 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 (§A.2) describes OPENORCA and MOPENORCA.

Fine-tuning Technique. We perform full fine-tuning in `bf16` precision to study the impact of multilingual post-training in Phase 2 on English ability. This setting allows us to fully leverage model capacity, which may not be attainable under parameter-efficient fine-tuning methods (Aggarwal et al., 2024; Panda et al., 2024). To mitigate the resulting degradation in English performance, §5 introduces a heuristic-based layer freezing strategy that selectively freezes a subset of layers while fine-tuning the remainder. All experiments are conducted using *Axolotl*[3], an open-source framework for large-scale LLM fine-tuning.

Evaluation Tasks. To quantify an LLM's english ability, we evaluate Phase 1 and Phase 2 models on two instruction-following tasks (i) `IFEval` (Zhou et al., 2023) and (ii) `Alpaca Eval` (Li et al., 2023), (iii) `MMLU` (Hendrycks et al., 2021) for problem-solving, (iv) `HellaSwag` (Zellers et al., 2019) for commonsense reasoning ability, and (v) `XLSUM_en` (Hasan et al., 2021) for summarization. To quantify an LLM's multilingual ability, we evaluate our fine-tuned models on three benchmark datasets comprising two multilingual generative tasks: question answering (`MLQA` (Lewis et al., 2019) & `XQuAD` (Artetxe et al., 2019)), instruction-following (`GMMLU` (Singh et al., 2024a)), and summarization (`XLSUM` (Hasan et al., 2021)).

Our evaluation suite spans diverse linguistic phenomena and reasoning skills to comprehensively assess language understanding, generation, and problem-solving. It also incorporates parallel benchmarks for English and multilingual ability, enabling direct cross-lingual comparison. Further details are provided in §A.3.

To evaluate our models on EA and MA, we use *LM-Evaluation-Harness*[4], which is a unified framework for zero/few-shot evaluations of LLMs. For both English and multilingual ability, we use **zero-shot** evaluation. For additional details on the training setup, code, and evaluation tasks, refer to §A.

4.2 Results

We compare the English and Multilingual ability of MISTRAL-7B and LLAMA-3-8B continually fine-tuned models on different phase-wise datasets[5]. Table 1 presents the results for English Ability (EA), while Table 2 presents the results for Multilingual Ability (MA). Table 2 reports the average score across languages. We provide language-specific scores and results when the phases are reversed (e.g., MULTIALPACA-ALPACA) in §B and §C.

Comparison with Mixture. From Tables 1 & 2: for Mixture, the mean of EA and MA scores for MISTRAL-7B fine-tuned on ALPACA-MULTIALPACA is 0.325, and 0.312 for LLAMA-3-8B. The corresponding two-phase mean score is 0.355 for MISTRAL-7B and 0.346 for LLAMA-3-8B. Our two-phase CFT setup is more effective than Mixture, for approximately the same number of training steps.

Discussion. From Table 1, 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

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

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

[5] 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.

Two-phase Continual Fine-tuning														
Model	**Phase 1 (P1) Dataset**	**Phase 2 (P2) Dataset**	IFEval (↑)		Alpaca Eval (↑)		MMLU (↑)		HellaSwag (↑)		XLSUM_en (↑)		**Average**	
			P1	**P2**	**P1**	**P2**	**P1**	**P2**	**P1**	**P2**	**P1**	**P2**	**P1**	**P2**
MISTRAL-7B	ALPACA		0.364	0.395	0.12	0.16	0.552	0.573	0.581	0.616	0.10	0.11	0.343	0.371
	Instruct	MULTI	0.550	0.462	0.35	0.15	0.575	0.533	0.641	0.416	0.13	0.10	0.449	0.332
LLAMA-3-8B	ALPACA	ALPACA	0.277	0.326	0.10	0.11	0.231	0.242	0.556	0.567	0.07	0.08	0.247	0.265
	Instruct		0.735	0.182	0.14	0.10	0.340	0.239	0.533	0.278	0.11	0.09	0.372	0.178
Dataset Mixture														
Model	**Dataset Mixture**		IFEval (↑)		Alpaca Eval (↑)		MMLU (↑)		HellaSwag (↑)		XLSUM_en (↑)		**Average**	
MISTRAL-7B	ALPACA MULTIALPACA		0.394		0.23		0.538		0.602		0.09		0.371	
LLAMA-3-8B	ALPACA MULTIALPACA		0.363		0.07		0.598		0.602		0.04		0.335	

Table 1: **English Ability results for two-phase Continual Fine-tuning (CFT).** When the phase-wise datasets are similar (Definition 1 and Definition 2), English 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 English Ability post Phase 2 (P2) fine-tuning (denote with red). We also provide numbers for dataset mixture – when the models are fine-tuned simultaneously on the Phase 1 and Phase 2 datasets.

Two-phase Continual Fine-tuning												
Model	**Phase 1 (P1) Dataset**	**Phase 2 (P2) Dataset**	MLQA (↑)		XLSUM (↑)		XQuAD (↑)		GMMLU (↑)		**Average**	
			P1	**P2**	**P1**	**P2**	**P1**	**P2**	**P1**	**P2**	**P1**	**P2**
MISTRAL-7B	ALPACA		0.229	0.288	0.012	0.060	0.290	0.602	0.42	0.40	0.238	0.338
	Instruct	MULTI	0.246	0.307	0.012	0.033	0.351	0.436	0.44	0.43	0.262	0.302
LLAMA-3-8B	ALPACA	ALPACA	0.438	0.597	0.033	0.034	0.586	0.737	0.53	0.34	0.397	0.427
	Instruct		0.609	0.321	0.048	0.027	0.712	0.417	0.25	0.44	0.405	0.301
Dataset Mixture												
Model	**Dataset Mixture**		MLQA (↑)		XLSUM (↑)		XQuAD (↑)		GMMLU (↑)		**Average**	
MISTRAL-7B	ALPACA MULTIALPACA		0.406		0.079		0.217		0.41		0.278	
LLAMA-3-8B	ALPACA MULTIALPACA		0.480		0.040		0.139		0.50		0.289	

Table 2: **Multilingual Ability results for two-phase Continual Fine-tuning (CFT).** With green, we denote an improvement in Multilingual Ability post Phase 2 fine-tuning. Likewise, we denote a decline in Multilingual Ability with red. For MLQA and XQUAD we use F1 abstractive score, while for XLSUM we use ROUGE (Lin, 2004) score. We also provide numbers for dataset mixture – when the models are fine-tuned simultaneously on the Phase 1 and Phase 2 datasets.

in Phase 2. However, we see a jump in MISTRAL-7B's multilingual ability for the multilingual generative tasks (Table 2). That is, Phase 2 models fine-tuned on multilingual datasets show forgetting in English. However, for phase-wise datasets like ALPACA followed by MULTIALPACA, we see that Phase 2 models do not show a decline in English ability (Table 1). We also see a gain in these models' multilingual ability (Table 2).

Ablations. In Tables B2 & B3 (§B), we present results for OPENORCA-MOPENORCA phase-wise datasets. First, the "dataset mixture" again performs worse on average than CFT: 0.186 vs. 0.372 for MISTRAL-7B and 0.217 vs. 0.366 for LLAMA-3-8B. Second, for MISTRAL-7B, the average English ability of the Phase 2 model (over Phase 1's MISTRAL-7B-OPENORCA) marginally improves: 0.413 from 0.407. Whereas, for MISTRAL-7B-INSTRUCT, the average decline in English ability is significant: 0.30 from 0.450. Likewise, for LLAMA-3-8B, the average English ability for LLAMA-3-8B OPENORCA MOPENORCA sees an increase to 0.356 from 0.337. In contrast, for Instruct-MOPENORCA, the English ability significantly drops, from 0.372 to 0.138.

Observation. With Table 1, we see that our two-phase CFT setup for multilingual adaptation 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 English 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 instructions in different languages*. We hypothesize an LLM fine-tuned on either of these datasets learns the same instructions, and therefore, the second phase of CFT leads to lesser interfer-

Phase 1 Dataset	Phase 2 Dataset	DES (↑)
ALPACA	MULTIALPACA	0.924
	MOPENORCA	0.792
OPENORCA	MOPENORCA	0.953
	MULTIALPACA	0.774
MISTRAL-7B Instruct‡	MULTIALPACA	0.746

‡: Prepared using model responses on MTBENCH (Zheng et al., 2024)

Table 3: Quantifying Phase-wise Dataset Similarity using DES: higher the score, greater the dataset similarity.

Dataset D_2	Model Parameter Difference (↓)
ALPACA	0.29
Instruct	1.00
OPENORCA	0.55

Table 4: Quantifying Phase-wise Similarity using MPD: lower the score, greater the dataset similarity. Here, we fix MULTIALPACA as D_1 and θ_B as MISTRAL-7B.

ence in the representation space. That is, an LLM continually fine-tuned on ALPACA & MULTIALPACA preserves its English ability across phases. We next define two metrics that aim to quantify the instruction-specific similarity of two datasets.

4.3 Similarity of Phase-wise Datasets

Dataset Embedding Similarity (DES). To quantify whether two datasets are similar[6], we define DES that computes a similarity score using the dot product of the average representations (embeddings) 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 $f_{DES} : D \times D \rightarrow [0, 1]$*

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

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 (Feng et al., 2020). We compute DES by encoding 500 random samples from ALPACA, MULTIALPACA, OPENORCA, and MOPENORCA, and measure f_{DES} for each pair. Table 3 presents the numbers. For dataset pairs with similar datasets, we see a high DES score and relatively low scores for dissimilar datasets. DES captures the (pair-wise) variation in instruction similarity of these datasets.

Model Parameter Difference (MPD). Another method of quantifying the similarity of instructions 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 datasets, the combined shift by Θ_2 should be relatively lower, compared to the shift if D_1 & D_2 encode different intstructions. 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 $f_{MPD} : \Theta \times \Theta \rightarrow \mathbb{R}_{\geq 0}$ s.t.*

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

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 (see Table 4). MPD shows a similar trend to DES: for ALPACA and 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 English Ability

Setup. To explain the effect of similar phase-wise data sets on an LLM's EA, we look at model representations when parsing English. We feed MTBENCH (Zheng et al., 2024) to the models, a widely-used English benchmark for generalized instruction-following evaluation, 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.

t-SNE Visualization. Figure 1 depicts t-SNEs (van der Maaten and Hinton, 2008) for

[6]The CL-ML literature often defines task similarity via permutation tasks, emphasizing input-output transformations (Goldfarb et al., 2024). Whereas, we consider semantic and structural similarity in natural language instructions.

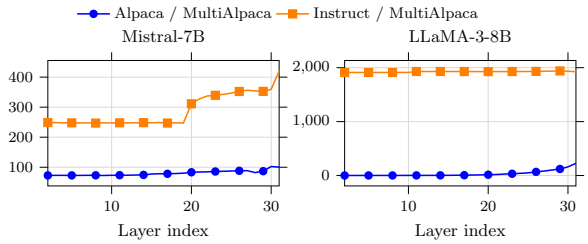


Figure 2: **Visualizing variance in model representations.** We plot $\|\Sigma_{\text{Phase 2}} - \Sigma_{\text{Phase 1}}\|_2$ across layers to measure representation shift. Dissimilar datasets (e.g., Instruct & MULTIALPACA) induce substantially larger variation than similar ones (e.g., ALPACA& MULTIALPACA). For LLAMA-3-8B, this variation is consistently high across layers, whereas for MISTRAL-7B, it is more localized and increases primarily in higher layers.

$X_{\text{MISTRAL-7B}}$ and $X_{\text{LLAMA-3-8B}}$ when these are continually fine-tuned on (i) ALPACA & MULTIALPACA and (ii) 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 separation between model representations results in increased weight interference during Phase 2 – leading to a decline in EA.

Visualizing Variance in Model Representations. Figure 1 provides an intuition for the correlation between phase-wise datasets and the decline in English ability. To further understand the layer-wise behavior of the hidden activations, similar to Chang et al. (2022), 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 $\bar{X}_\Theta = X_\Theta - \mu_\Theta$, where $\mu_\Theta = 1/l \sum_{i=1}^{l} X_\Theta^{(i)}$. 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 CFT

To mitigate EA decline, we explore two tailored CFT techniques[7]: Distribution Replay and Layer Freezing. In Distribution Replay, we study Generative Replay (GR), a new English data generation method inspired by dataset similarity and English ability (§4.2), and English Replay (ER), which replays parallel English data of Phase 2's distribution. In Layer Freezing (LF), we identify layers to freeze during Phase 2 fine-tuning using specific heuristics.

5.1 Distribution 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 (Shin et al., 2017). However, from §4.2, we do not see a decline in English ability if the phase-wise datasets encode similar instructions. Based on this, we use the Phase 1 model to generate responses, in English, from the English counterpart of the multilingual dataset used for fine-tuning in Phase 2. The intuition is that the generated dataset may bridge 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. We also fine-tune the models with a similar sized subset of the English counterpart with original responses[8]. We refer to this mitigating strategy as English Replay (ER_10).

5.2 Layer Freezing

Model regularization is an effective technique to mitigate the drop in the previous task's performance in continual learning (e.g., EWC (Kirkpatrick et al., 2017)). However, this is computationally inefficient as it requires using both the old

[7] In §F, we discuss how MAD-X (Pfeiffer et al., 2020) and other PEFT-based methods (Badola et al., 2023) are not suitable for our setting.

[8] This dataset may not be available for all multilingual datasets, such as Aya (Singh et al., 2024b). While instructions can be translated into English, translating responses is often impractical. Thus, ER is the best-case scenario for GR.

CFT Setup		English Ability (EA)						Multilingual Ability (MA)					Combined
	Mitigating Strategy	IFEval (↑)	Alpaca Eval (↑)	MMLU (↑)	HellaSwag (↑)	XLSUM_en (↑)	Avg (↑)	MLQA (↑)	XLSUM (↑)	XQUAD (↑)	GMMLU (↑)	Avg (↑)	Avg (↑)
MISTRAL-7B	–	0.462	0.15	0.533	0.416	0.10	0.332	0.307	0.033	0.436	0.430	0.302	0.317
	LF_H1	0.456	0.03	0.497	0.598	0.10	0.336	0.176	0.016	0.215	0.428	0.209	0.273
	LF_H2	0.364	0.12	0.364	0.504	0.12	0.294	0.213	0.014	0.442	0.384	0.263	0.279
	Spectrum	0.435	**0.24**	0.488	0.524	**0.13**	0.363	**0.317**	**0.083**	0.176	0.370	0.237	0.30
	GR_5	0.540	0.17	0.540	0.611	0.11	0.394	0.311	0.008	0.428	0.445	0.298	**0.346**
	GR_10	0.567	0.12	0.567	0.594	0.12	0.394	0.213	0.007	0.427	**0.450**	0.274	0.334
	ER_10	**0.593**	0.08	**0.580**	**0.635**	**0.13**	**0.404**	0.249	0.008	0.398	0.448	0.276	0.340
	LoRA	0.383	0.09	0.579	0.625	0.03	0.341	0.289	0.043	**0.518**	0.435	**0.321**	0.331
LLAMA-3-8B	–	0.182	0.10	0.239	0.278	0.09	0.178	0.321	0.030	0.417	0.440	0.302	0.240
	LF_H1	0.303	0.0	0.231	0.275	0.01	0.164	0.368	0.037	0.505	0.237	0.287	0.225
	LF_H2	0.380	0.06	0.485	0.525	0.08	0.306	0.400	0.038	0.505	0.338	0.320	0.313
	Spectrum	0.409	0.09	0.612	0.524	0.01	0.329	0.429	**0.056**	0.086	**0.472**	0.261	0.295
	GR_5	0.269	0.01	0.516	0.316	0.07	0.236	**0.437**	0.019	**0.593**	0.342	0.348	0.292
	GR_10	0.264	**0.12**	0.229	0.250	0.0	0.173	0.254	0.009	0.314	0.238	0.204	0.189
	ER_10	**0.420**	0.02	**0.603**	**0.561**	**0.12**	**0.345**	0.434	0.025	0.53	0.448	**0.359**	**0.352**
	LoRA	0.196	0.0	0.280	0.235	0.0	0.142	0.007	0.008	0.005	0.278	0.075	0.109

Table 5: **English Ability (EA) and Multilingual Ability (MA) results for our mitigating strategies.** These comprise Generative Replay (GR_5 & GR_10), English Replay (ER_10) and Layer Freezing (LF_H1, LF_H2 & Spectrum). We use LoRA (Hu et al., 2022) as a baseline strategy. For ER_10, we use the English dataset used in GR with original responses. *The Phase 1 dataset is Instruct for each row, while Phase 2 is* MULTIALPACA. The first row for both MISTRAL-7B and LLAMA-3-8B provides numbers for Instruct-MULTIALPACA (from Table 1 & 2).

and new sets of parameters. Instead, we use Layer Freezing (LF), a relatively efficient technique for use as a 'regularizer' to preserve English ability during Phase 2. We consider the following variations to select the set of layers to freeze. These variants allow us to study how different criteria for selecting frozen layers affect the trade-off between preserving English ability and enabling multilingual adaptation.

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 layers separately for Key, Query, and Value, for each attention head.

3. Spectrum (Hartford et al., 2024): freeze the "most informative" layers of the Phase 1 model based on their signal-to-noise ratio (refer to §D.1).

We present our results in Table 5 for both GR and LF. We define a **baseline** in which we use LoRA (Hu et al., 2022)[9] for continually fine-tuning in Phase 2. We perform LoRA fine-tuning with rank 64 and quantisation bfloat16.

[9]Parameter efficient techniques like LoRA (Hu et al., 2022) are also widely used to efficiently fine-tune LLMs on multilingual data. However, such techniques also show *forgetting* on English (Aggarwal et al., 2024) after Phase 2.

5.3 Results Discussion

From Table 5, we see that GR, ER, and LF mitigate the decline in EA and also show gains in MA.

Distribution Replay. ER_10 demonstrates the best performance in both English and combined ability, with EA scores of 0.404 for MISTRAL-7B and 0.345 for LLAMA-3-8B, and the best combined average. GR_5 also excels in multilingual tasks, performing similar to ER_10: 0.298 vs. 0.276 for MISTRAL-7B and 0.348 vs. 0.359 for LLAMA-3-8B. GR_5 also performs reasonably well on English tasks, achieving scores of 0.394 and 0.236 for MISTRAL-7B and LLAMA-3-8B, respectively, making it a competitive strategy.

Layer Freezing. Compared to ER and GR, LF_H1, LF_H2, and Spectrum show mixed results. LF_H2 performs better than LF_H1. Spectrum's EA scores are better than LF_H1 and LF_H2, but suffers from lower multilingual numbers.

Additional Discussion & Results. In §D.3, we also present EA and MA results for MISTRAL-7B Instruct-MOPENORCA for our mitigating strategies. Here, LF, particularly Spectrum, performs better than the other strategies. Furthermore, in §D.4, we analyze the computational cost of these strategies over the baseline CFT setup.

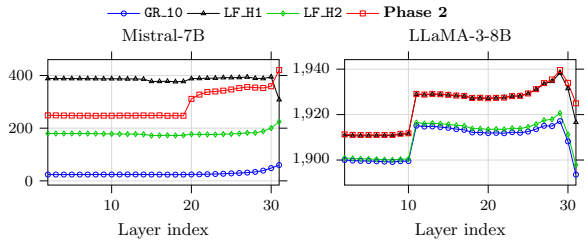


Figure 3: **Layer-wise covariance shift under mitigation strategies in Phase 2 CFT.** We plot $\|\Sigma_{\text{Phase 2}} - \Sigma_{\text{Phase 1}}\|_2$ across layers. The vanilla Phase 2 model (Figure 2) exhibits a sharp increase in drift in higher layers. Replay (GR) substantially reduces the magnitude of this drift across all layers, while freezing (LF_H2) confines it to a smaller subset of layers.

5.4 Controlling Representation Drift under Mitigation

Building on Figure 2, we examine how mitigation strategies affect representation drift relative to the vanilla Phase 2 baseline (MISTRAL-7B Instruct → MULTIALPACA). As in §4.4, we measure drift as $\|\Sigma_{\text{Phase 2}} - \Sigma_{\text{Phase 1}}\|_2$, where larger values indicate greater deviation from Phase 1 representations.

Consistent with earlier observations, the mitigation strategies explicitly designed to curb representational change (LF & GR) exhibit lower drift than the baseline model across layers. Replay-based methods (GR) reduce the overall magnitude of drift while preserving its layer-wise structure, whereas freezing-based methods (LF_H2) constrain the drift to a subset of layers.

Notably, generative replay remains closest to the Phase 1 representation geometry among all strategies. This closer alignment correlates with improved task and language performance compared to the vanilla Phase 2 model (Table 1 and Table 2). These results reinforce our central hypothesis: in two-phase CFT with dissimilar datasets, representation shift is the primary driver of English degradation, and explicitly controlling this shift preserves alignment with Phase 1 representations, leading to improved English retention without sacrificing multilingual gains.

6 Conclusion & Future Work

We study the role of dataset similarity in a novel two-phase continual fine-tuning (CFT) for multilingual adaptation. Across MISTRAL-7B and LLAMA-3-8B, we show that alignment between Phase 1 and Phase 2 datasets is critical: similar datasets preserve English ability while improving multilingual performance, whereas misaligned datasets lead to degradation. We identify representation shift as the underlying mechanism driving this behavior. In particular, misaligned fine-tuning induces large covariance shifts in hidden representations, concentrated in higher layers. We show that mitigation strategies such as generative replay and layer freezing improve English retention by controlling this drift, either by globally reducing its magnitude or by constraining it to specific layers, while preserving multilingual gains.

Future Work. Our results suggest several directions for future research. First, designing adaptive mitigation strategies that dynamically control representation drift during training could provide stronger and more efficient trade-offs than fixed heuristics (Zhang et al., 2026). Second, developing principled measures of instruction-level dataset similarity may enable better prediction and prevention of degradation. Finally, exploring parameter-efficient approaches that explicitly regularize representation geometry could reduce the computational overhead of current methods while maintaining their effectiveness.

7 Limitations

Our study has several limitations. First, we rely on DES and MPD as proxies for dataset similarity; while effective in our setting, these metrics may not capture all nuances of instruction-level similarity. Second, our experiments are limited to MISTRAL-7B and LLAMA-3-8B, and the observed trends may not fully generalize to models with different architectures, scales, or training paradigms. Similarly, the effectiveness of our mitigation strategies (i.e., replay and layer freezing) may vary across datasets and model families. Third, our experimental setup is constrained by computational resources. We were unable to explore larger models, longer training schedules, or broader hyperparameter sweeps, which may further influence the observed trade-offs. Fourth, some mitigation strategies impose additional requirements. In particular, the best-performing method, ER_10, assumes access to parallel data, which may not always be available in practice. Finally, our evaluation focuses on specific benchmarks for task and language ability. While these provide controlled comparisons, they may not fully capture real-world performance across diverse applications.

References

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

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267.

Sanchit Ahuja, Divyanshu Aggarwal, Varun Gumma, Ishaan Watts, Ashutosh Sathe, Millicent Ochieng, Rishav Hada, Prachi Jain, Mohamed Ahmed, Kalika Bali, et al. 2024a. Megaverse: Benchmarking large language models across languages, modalities, models and tasks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2598–2637.

Sanchit Ahuja, Kumar Tanmay, Hardik Hansrajbhai Chauhan, Barun Patra, Kriti Aggarwal, Luciano Del Corro, Arindam Mitra, Tejas Indulal Dhamecha, Ahmed Awadallah, Monojit Choudhary, Vishrav Chaudhary, and Sunayana Sitaram. 2024b. sphinx: Sample efficient multilingual instruction fine-tuning through n-shot guided prompting. *Preprint*, arXiv:2407.09879.

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

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

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

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. 2025. Nested learning: The illusion of deep learning architectures. *arXiv preprint arXiv:2512.24695*.

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

Samuel Cahyawijaya, Holy Lovenia, Tiezheng Yu, Willy Chung, and Pascale Fung. 2023. 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.

Salvador Carrión and Francisco Casacuberta. 2022. 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.

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

Hongyang Chen, Zhongwu Sun, Hongfei Ye, Kunchi Li, and Xuemin Lin. 2026. Continual learning in large language models: Methods, challenges, and opportunities. *arXiv preprint arXiv:2603.12658*.

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

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

Tejas Dhamecha, Rudra Murthy, Samarth Bharadwaj, Karthik Sankaranarayanan, and Pushpak Bhattacharyya. 2021. 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.

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. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *Preprint*, arXiv:2007.01852.

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

Daniel Goldfarb, Itay Evron, Nir Weinberger, Daniel Soudry, and PAul HAnd. 2024. The joint effect of task similarity and overparameterization on catastrophic forgetting—an analytical model. In *The Twelfth International Conference on Learning Representations*.

Zheng Gong, Kun Zhou, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2022. 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.

Changhao Guan, Chao Huang, Hongliang Li, You Li, Ning Cheng, Zihe Liu, Yufeng Chen, Jinan Xu, and Jian Liu. 2025. Multi-stage LLM fine-tuning with a continual learning setting. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5484–5498, Albuquerque, New Mexico. Association for Computational Linguistics.

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

Eric Hartford, Lucas Atkins, Fernando Fernandes Neto, and David Golchinfar. 2024. Spectrum: Targeted training on signal to noise ratio. *arXiv preprint arXiv:2406.06623*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. 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. Association for Computational Linguistics.

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

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

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

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. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. 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.

Jiazheng Kang, Le Huang, Cheng Hou, Zhe Zhao, Zhenxiang Yan, and Ting Bai. 2025. Self-evolving llms via continual instruction tuning. *arXiv preprint arXiv:2509.18133*.

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

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.

Séamus Lankford, Haithem Afli, and Andy Way. 2023. adaptmllm: Fine-tuning multilingual language models on low-resource languages with integrated llm playgrounds. *Information*, 14(12):638.

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

Chong Li, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2024a. Improving in-context learning of multilingual generative language models with cross-lingual alignment. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8051–8069.

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

Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ali Payani, Ninghao Liu, and Mengnan Du. 2024b. Language ranker: A metric for quantifying llm performance across high and low-resource languages. *Preprint*, arXiv:2404.11553.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, Andrea Madotto, and Pascale Fung. 2021. Preserving cross-linguality of pre-trained models via continual learning. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 64–71, Online. Association for Computational Linguistics.

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

Vladimir Alexandrovich Marchenko and Leonid Andreevich Pastur. 1967. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114(4):507–536.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

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

Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, et al. 2023. Seallms–large language models for southeast asia. *arXiv preprint arXiv:2312.00738*.

Ashwinee Panda, Berivan Isik, Xiangyu Qi, Sanmi Koyejo, Tsachy Weissman, and Prateek Mittal. 2024. Lottery ticket adaptation: Mitigating destructive interference in llms. *Preprint*, arXiv:2406.16797.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. 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.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *Preprint*, arXiv:2005.00052.

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

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

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

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *Preprint*, arXiv:1705.08690.

Amal Shiyas. 2023. Microsoft research project helps languages survive — and thrive.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David I Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, et al. 2024a. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation. *arXiv preprint arXiv:2412.03304*.

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. 2024b. Aya dataset: An open-access collection for multilingual instruction tuning. *Preprint*, arXiv:2402.06619.

Vaibhav Singh, Amrith Krishna, Karthika NJ, and Ganesh Ramakrishnan. 2024c. A three-pronged approach to cross-lingual adaptation with multilingual llms. *arXiv preprint arXiv:2406.17377*.

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

Regina Ta. 2023. How language gaps constrain generative ai development. Brookings Institution, Online Article.

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

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

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. 2023. Polylm: An open source polyglot large language model. *Preprint*, arXiv:2307.06018.

Genta Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preoţiuc-Pietro. 2023. Overcoming catastrophic forgetting in massively multilingual continual learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 768–777.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura,

Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Tongtong Wu, Trang Vu, Linhao Luo, and Gholamreza Haffari. 2025. Continual learning of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 16–17, Suzhou, China. Association for Computational Linguistics.

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

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

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

Yanchun Zhang and Guandong Xu. 2009. *Singular Value Decomposition*, pages 2657–2658. Springer US, Boston, MA.

Zhixin Zhang, Zeming Wei, and Meng Sun. 2026. Dynamic orthogonal continual fine-tuning for mitigating catastrophic forgetting of LLMs. In *4th Deployable AI Workshop*.

Yiran Zhao, Wenxuan Zhang, Huiming Wang, Kenji Kawaguchi, and Lidong Bing. 2024. Adamergex: Cross-lingual transfer with large language models via adaptive adapter merging. *Preprint*, arXiv:2402.18913.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

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

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
Optimizer	AdamW (Loshchilov and Hutter, 2019)
Weight Decay	0

Table A1: Hyperparameters for our Two-phase Continual Fine-tuning (CFT)

A.2 Fine-tuning Datasets

OPENORCA is an English-only self instruct dataset, created to best mimic the ORCA dataset (Mukherjee et al., 2023), which is not publicly available. To create the multilingual version of OPENORCA, namely MOPENORCA, we follow Ahuja et al. (2024b) to generate selective translations for a subset of OPENORCA. The subset contains 50k samples from the OPENORCA dataset and we selectively translate them to 11 languages which are also in MULTIALPACA. In total, we generate 550k examples for all languages.

A.3 Evaluation Tasks

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

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

1. `IFEval` (Zhou et al., 2023): 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` (Li et al., 2023): This is an LLM-based automatic evaluator for instruction following models, to measure task ability. Like Aggarwal et al. (2024), we evaluate our CFT models against *text-davinci-003* responses on 800 instructions and use GPT4 (*gpt-4-32k)* as the evaluator.
3. `MMLU` (Hendrycks et al., 2021): 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` (Zellers et al., 2019): This is a popular benchmark to evaluate the commonsense reasoning ability of an LLM. `HellaSwag`'s test split contains 10k samples in total.

Multilingual Ability (MA). 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 summarization.

- **Question Answering:** `MLQA` (Lewis et al., 2019) contains 5k extractive question-answering instances in 7 languages. The `XQuAD` dataset (Artetxe et al., 2019) consists of a subset of 240 paragraphs and 1190 question-answer pairs across 11 languages.
- **Summarisation:** `XLSUM` (Hasan et al., 2021) spans 45 languages, and we evaluate our models in Arabic, Chinese-Simplified, English, French, Hindi, Japanese, and Spanish.

B Evaluating Multilingual Ability for Continual Fine-tuning

English Ability. Table B2 present the english ability numbers of our ablations on the OPENORCA-MOPENORCAand Instruct-MOPENORCAdatasets using MISTRAL-7B and LLAMA-3-8B models. When the datasets are pairwise not similar, i.e., Instruct-MOPENORCA, MISTRAL-7B shows a significant decline in the *average* English ability, from 0.450 in Phase 1 to 0.30 in Phase 2. Likewise, LLAMA-3-8B also experiences a decrease, dropping from 0.372 to 0.138 on average.

In contrast, when the pairwise datasets are similar, i.e., OPENORCA and MOPENORCA, MISTRAL-7B sees a marginal increase between the phases $(0.407 \rightarrow 0.413)$, on average. LLAMA-3-8B's performance sees an improvement in the average English ability, from 0.337 to 0.356.

Multilingual Ability. Table B3 tabulates the results for multilingual ability. We see an improvement in the *average* multilingual ability for the OPENORCA-MOPENORCA dataset pair for LLAMA-3-8B. For MISTRAL-7B, there is a

Two-phase Continual Fine-tuning														
Model	**Phase 1 (P1) Dataset**	**Phase 2 (P2) Dataset**	IFEval (↑)		Alpaca Eval (↑)		MMLU (↑)		HellaSwag (↑)		XLSUM_en (↑)		**Average**	
			P1	**P2**	**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	0.02	0.12	0.407	0.413
	Instruct	MOPENORCA	0.550	0.426	0.35	0.06	0.575	0.507	0.641	0.509	0.13	0.0	0.450	0.30
LLAMA-3-8B	OPENORCA	MOPENORCA	0.377	0.425	0.09	0.07	0.579	0.599	0.571	0.564	0.07	0.12	0.337	0.356
	Instruct	MOPENORCA	0.735	0.205	0.14	0.0	0.340	0.236	0.533	0.250	0.11	0.0	0.372	0.138

Dataset Mixture						
Model	**Dataset Mixture**	IFEval (↑)	Alpaca Eval (↑)	MMLU (↑)	HellaSwag (↑)	XLSUM_en (↑)	**Average**
MISTRAL-7B	OPENORCA MOPENORCA	0.228	0.035	0.284	0.444	0.02	0.202
LLAMA-3-8B	OPENORCA MOPENORCA	0.248	0.072	0.484	0.473	0.0	0.255

Table B2: English 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.

Two-phase Continual Fine-tuning														
Model	**Phase 1 (P1) Dataset**	**Phase 2 (P2) Dataset**	MLQA (↑)		XLSUM (↑)		XQuAD (↑)		GMMLU (↑)		**Average**	
			P1	**P2**	**P1**	**P2**	**P1**	**P2**	**P1**	**P2**	**P1**	**P2**
MISTRAL-7B	OPENORCA	MOPENORCA	0.435	0.360	0.007	0.008	0.556	0.643	0.433	0.312	0.358	0.331
	Instruct	MOPENORCA	0.246	0.155	0.012	0.040	0.351	0.323	0.440	0.279	0.262	0.20
LLAMA-3-8B	OPENORCA	MOPENORCA	0.401	0.453	0.017	0.006	0.499	0.531	0.242	0.513	0.290	0.376
	Instruct	MOPENORCA	0.609	0.604	0.048	0.048	0.712	0.713	0.250	0.233	0.405	0.40

Dataset Mixture						
Model	**Dataset Mixture**	MLQA (↑)	XLSUM (↑)	XQuAD (↑)	GMMLU (↑)	**Average**
MISTRAL-7B	OPENORCA MOPENORCA	0.201	0.128	0.071	0.277	0.169
LLAMA-3-8B	OPENORCA MOPENORCA	0.224	0.034	0.091	0.364	0.178

Table B3: Multilingual Ability results for two-phase Continual Fine-tuning (CFT). With green, we highlight an increase in a model's Multilingual ability post Phase 2 fine-tuning. Likewise, red highlights a decline in a model's Multilingual ability.

marginal drop ($0.358 \rightarrow 0.331$). For Instruct-MOPENORCA, with LLAMA-3-8B, the average multilingual ability is virtually the same across tasks (0.405 vs. 0.40). However, for MISTRAL-7B, we see a slight drop in the average language ability, driven primarily due to a decline in performance for MLQA.

Furthermore, Table B6, Table B7, and Table B8 present the language-specific results for MLQA, XLSUM, and XQuAD, respectively.

C Reverse Order CFT Result Analysis

In Tables B4 and B5, we reverse the order of Phase 1 and Phase 2 datasets, first fine-tuning on the multilingual dataset followed by its English counterpart. For MISTRAL-7B (MULTIALPACA-ALPACA), the average performance is 0.226, and for LLAMA-3-8B (MULTIALPACA-ALPACA), it is 0.259. Compared to the mixture setting and the ALPACA-MULTIALPACA configuration (§4), we observe that English ability benefits from multilingual fine-tuning in Phase 1, resulting in performance comparable to the data mixture setting. However, fine-tuning on English data in Phase 2 leads to a drastic drop in multilingual ability, yielding worse results than both the mixture setting and the two-phase setup considered in the main paper.

D Mitigating Strategies

Here, we provide additional details on Spectrum (Hartford et al., 2024). We then visualize the impact of our mitigating strategies on the variance in model representations. Lastly, we ablate our findings for the Instruct-MOPENORCA phase-wise datasets.

D.1 Spectrum

Spectrum (Hartford et al., 2024) is a layer-freezing technique that optimizes the fine-tuning of LLMs by selecting layers based on their signal-to-noise ratio (SNR). We use Spectrum as a heuristic for layer-freezing; that is, the layers identified as "important" by Spectrum are frozen during Phase 2 fine-tuning. A layer is important based on its signal-to-noise (SNR) ratio. In the following, we elaborate on how Spectrum computes SNR.

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	MULTIALPACA	ALPACA	0.245	0.290	0.120	0.114	0.528	0.430	0.476	0.510	0.342	0.336
LLAMA-3-8B	MULTIALPACA	ALPACA	0.245	0.340	0.038	0.065	0.570	0.540	0.577	0.590	0.357	0.384
MISTRAL-7B	MOPENORCA	OPENORCA	0.190	0.310	0.091	0.055	0.410	0.490	0.520	0.510	0.303	0.341
LLAMA-3-8B	MOPENORCA	OPENORCA	0.314	0.340	0.0	0.0	0.530	0.540	0.522	0.590	0.342	0.368

Table B4: English Ability results for two-phase Continual Fine-tuning (CFT)

Model	Phase 1 Dataset	Phase 2 Dataset	MLQA (↑)		XLSUM (↑)		XQuAD (↑)		Average	
			Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2
MISTRAL-7B	MULTIALPACA	ALPACA	0.122	0.230	0.021	0.030	0.122	0.090	0.088	0.116
LLAMA-3-8B	MULTIALPACA	ALPACA	0.363	0.340	0.048	0.040	0.058	0.030	0.157	0.134
MISTRAL-7B	MOPENORCA	OPENORCA	0.165	0.160	0.077	0.070	0.140	0.180	0.127	0.137
LLAMA-3-8B	MOPENORCA	OPENORCA	0.057	0.0	0.038	0.0	0.047	0.0	0.047	0.0

Table B5: Multilingual Ability results for two-phase Continual Fine-tuning (CFT)

Model	Phase 1 Dataset	Phase 2 Dataset	MLQA											
			Phase 1						Phase 2					
			ar	de	es	hi	vi	zh	ar	de	es	hi	vi	zh
MISTRAL-7B	ALPACA Instruct	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
			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 Instruct		0.320	0.538	0.563	0.438	0.611	0.155	0.552	0.672	0.765	0.573	0.784	0.237
			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 Instruct	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
			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 Instruct		0.262	0.545	0.565	0.369	0.568	0.099	0.437	0.549	0.622	0.462	0.625	0.024
			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 B6: MLQA: Language Ability results for two-phase Continual Fine-tuning (CFT).

Marchenko-Pastur distribution. The Marchenko-Pastur distribution (Marchenko and Pastur, 1967) is given by:

$$\rho(\lambda) = \frac{1}{2\pi\sigma^2} \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{\lambda},$$

where

$$\lambda_\pm = \sigma^2(1 \pm \sqrt{Q})^2,$$

and $Q = \frac{N}{M}$, with N and M being the dimensions of a random matrix W, and σ^2 representing the variance of the entries in W.

SNR. Let $W \in \mathbb{R}^{N \times M}$ be the weight matrix of a given layer. The empirical spectral density of W is analyzed by comparing its eigenvalue distribution of $1/N \cdot W^T W$ against the theoretical Marchenko-Pastur distribution. Deviations from this distribution indicate the presence of significant signal components. We get,

$$\lambda_\pm = \sigma^2 \left(1 \pm \sqrt{\frac{M}{N}}\right)^2,$$

where $\lambda_\pm$ are the largest and smallest eigenvalues and σ the standard deviation. This implies the bounds of singular values of W as:

$$\epsilon_\pm = \frac{1}{\sqrt{N}} \sigma \left(1 \pm \sqrt{\frac{M}{N}}\right) \quad (1)$$

By evaluating how the singular values of W distribute relative to $\epsilon_\pm$, Spectrum assesses the SNR of each layer, as defined next.

Ratio (Hartford et al., 2024). Specifically, the SNR value of a weight matrix is,

$$\text{SNR} = \frac{\sum_{k|\sigma_k > \epsilon} \sigma_k}{\sum_{k|\sigma_n < \epsilon} \sigma_n}$$

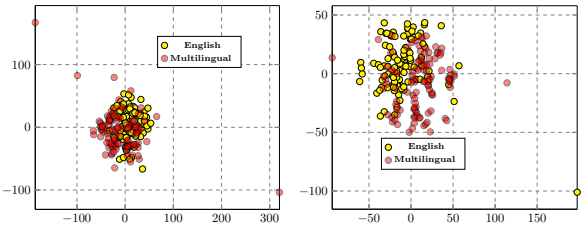
Here, ϵ separates signal from noisy singular values. Layers with singular values significantly exceeding ϵ_+ have a high SNR, indicating a substantial presence of informative signal components.

Measuring the Ratio (Hartford et al., 2024). Having defined all ingredients above, Spectrum now computes each layer's SNRs. To do this, it first computes SVD (Zhang and Xu, 2009) of the the layer's weight matrix, calculates the SNR and normalizes it by the highest singular value. Eq. 1 gives the noise threshold.

Now, Spectrum selects layers with higher SNRs, where the number of layers selected is a hyperparameter. Similar to Hartford et al. (2024), for our experiments, we select the top-50% of layers in each module.

D.2 LLaMA-3-8B Doesn't Show Consistent Improvement with our Mitigation Strategies

From Table 5, while both GR and LF improve on the baseline LLaMA-3-8B-Instruct MultiAlpaca, the gains in task and multilingual ability are not comparable to LLaMA-3-8B-Instruct.



(a) Mistral-7B (b) LLaMA-3-8B

Figure D4: Demonstrating extent of cross-lingual transfer in Mistral-7B and LLaMA-3-8B on a parallel dataset prepared by subsampling Flores (Costa-jussà et al., 2022). We find that the English activation cluster for LLaMA-3-8B is separated from the multilingual cluster, compared to Mistral-7B.

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 (Costa-jussà et al., 2022). In Figure D4, 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 MA 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.

D.3 Additional Ablations

We also present the impact of our mitigating strategies for the Instruct-mOpenOrca phase-wise datasets on Mistral-7B. Table D9 presents these results.

We see that LF_H2 achieves moderate success, especially in maintaining the language ability for MLQA (0.258) and XQUAD (0.527). However, task ability shows some decline (e.g., IFEval (0.401) and Alpaca Eval (0.048)), compared to the baseline. Furthermore, GR_5 results in lower task ability (IFEval = 0.281), while GR_10 performs slightly better in task ability (e.g., MMLU = 0.483, HellaSwag = 0.494). Among the baselines, ER_10 performs similarly to the generative replay strategies, with modest improvements in task ability (e.g., IFEval = 0.367, MMLU = 0.479), but still struggles in language ability. Perhaps LoRA shows the best overall performance among the strategies for maintaining task ability (e.g., IFEval = 0.587, MMLU = 0.567, HellaSwag = 0.591) with reasonable retention of language ability (e.g., XQUAD = 0.354).

These results show that no single strategy is perfect, and future work may need to combine these strategies or develop new approaches to address the balance between task and language ability retention across phases.

D.4 Compute Analysis

The computational overhead of replay arises from an increase in the effective training set size . In ER_10, each epoch includes the original Phase 2 data along with an additional 10% replay buffer from Phase 1, resulting in a proportional increase in the number of training tokens and hence compute. Similarly, GR_5 augments each epoch with 5% generated replay data, leading to a smaller overhead.

In contrast, layer freezing (LF) significantly reduces computation by updating only a subset of model parameters. By freezing 50% of the layers, LF yields a corresponding reduction in training cost while still achieving a reasonable trade-off between English retention and multilingual adaptation.

E Resources Used

We conduct all experiments on 4 NVIDIA A100 GPUs (80GB each) with a 96-core AMD CPU. A single fine-tuning run on MULTIALPACA takes approximately 4 hours, while training on MOPENORCA requires around 12 hours. The models used in our experiments, along with their corresponding checkpoints and licenses, are listed below:

- LLAMA-3-8B: https://huggingface.co/meta-llama/Meta-Llama-3-8B **License:** LLaMA 3
- MISTRAL-7B: https://huggingface.co/mistralai/Mistral-7B-v0.1 **License:** Apache-2.0

F Existing Mitigating Strategies

We now discuss why MAD-X (Pfeiffer et al., 2020) and other PEFT-based methods (Badola et al., 2023) are not well-suited to our setting.

- **Parameter-Efficient Fine-tuning for Robust Continual Multilingual Learning** (Badola et al., 2023): This work considers a fixed downstream task (text classification) that is incrementally extended to new languages, mitigating forgetting via task- and language-specific adapters. Their approach assumes continued access to prior task data and the ability to maintain multiple adapter checkpoints. In contrast, our setting involves end-to-end instruction fine-tuning over a heterogeneous mixture of tasks. We do not assume access to Phase 1 data during Phase 2, nor do we maintain multiple parameter sets due to the memory overhead of storing additional model weights (§3).
- **MAD-X** (Pfeiffer et al., 2020): MAD-X introduces bottleneck adapters within encoder-only or encoder–decoder Transformer architectures to enable cross-lingual transfer. However, recent work (Zhao et al., 2024) shows that such adapters do not integrate cleanly with decoder-only LLMs (e.g., Mistral-7B, LLaMA-3-8B), which form the basis of our experiments. As our models lack an encoder stack, the routing and merging mechanisms central to MAD-X are not directly applicable.
- **Preserving Cross-Linguality via Continual Learning** (Liu et al., 2021): This method extends Gradient Episodic Memory by storing exemplars (or their gradients) from previous phases and replaying them during training. Our setup (§3) explicitly disallows (i) reusing Phase 1 data and (ii) maintaining Phase 1 representations or weights during Phase 2, as these incur significant memory overhead and may violate practical data-sharing constraints.

Model	Phase 1 Dataset	Phase 2 Dataset	XLSUM											
			Phase 1						Phase 2					
			Arabic	Chinese_simplified	french	Hindi	Japanese	Spanish	Arabic	Chinese_simplified	french	Hindi	Japanese	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 B7: XLSUM: Language Ability results for two-phase Continual Fine-tuning (CFT).

Model	Phase 1 Dataset	Phase 2 Dataset	XQuAD																				
			Phase 1											Phase 2									
			ar	de	el	es	hi	ro	ru	th	tr	vi	zh	ar	de	el	es	hi	ro	ru	th	tr	vi	zh
MISTRAL-7B	ALPACA	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
	Instruct		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		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
	Instruct		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	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
	Instruct		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		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
	Instruct		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 B8: XQuAD: Language Ability results for two-phase Continual Fine-tuning (CFT).

CFT Setup			Task Ability					Language Ability				Overall
Model	**Phase 2 Dataset**	**Mitigating Strategy**	**IFEval**	**ALPACA Eval**	**MMLU**	**HellaSwag**	**Avg**	**MLQA**	**XLSum**	**XQUAD**	**Avg**	**Avg**
MISTRAL-7B	MOPENORCA	–	0.426	0.060	0.507	0.509	0.376	0.155	0.040	0.323	0.173	0.275
		LF_H2	0.401	0.048	0.518	0.487	0.364	0.258	0.060	**0.527**	**0.282**	0.323
		Spectrum	0.442	**0.158**	0.508	**0.616**	0.431	**0.387**	**0.086**	0.201	0.225	**0.328**
		GR_5	0.281	0.027	0.478	0.495	0.320	0.167	0.042	0.305	0.171	0.246
		GR_10	0.305	0.013	0.483	0.494	0.324	0.150	0.038	0.238	0.142	0.233
		ER_10	0.367	0.025	0.479	0.493	0.341	0.157	0.042	0.305	0.168	0.255
		LoRA	**0.587**	0.130	**0.567**	0.591	**0.469**	0.167	0.027	0.354	0.183	0.326

Table D9: English and Multilingual Ability results for our mitigating strategies, Generative Replay (GR_5 & GR_10), English Replay (ER_10) and Layer Freezing (LF_H1, LF_H2 & Spectrum). We use LoRA (Hu et al., 2022) as a baseline strategy. For ER_10, we use the English dataset used in GR with original responses. *The Phase 1 dataset is Instruct for each row.* The first row provides MISTRAL-7B numbers for Instruct-MOPENORCA (from Table B2).