The Unreasonable Effectiveness of Model Merging for Cross-Lingual Transfer in LLMs

Lucas Bandarkar* **Nanyun Peng**
University of California, Los Angeles

Abstract

Large language models (LLMs) still struggle across tasks outside of high-resource languages. In this work, we investigate cross-lingual transfer to lower-resource languages where task-specific post-training data is scarce. Building on prior work, we first validate that the subsets of model parameters that matter most for mathematical reasoning and multilingual capabilities are distinctly non-overlapping. To exploit this implicit separability between task and target language parameterization, we develop and analyze numerous *modular* frameworks to improve the *composition* of the two during fine-tuning. These methods generally employ freezing parameters or post hoc model merging to assign math and language improvement to different key parts of the LLM. In the absence of in-language math data, we demonstrate that the modular approaches successfully improve upon baselines across three languages, four models, and two fine-tuning paradigms (full and LoRA). Furthermore, we identify the most consistently successful modular method to be fine-tuning separate language and math experts and model merging via Layer-Swapping (Bandarkar et al., 2025a), somewhat surprisingly. We offer possible explanations for this result via recent works on the linearity of task vectors. We further explain this by empirically showing that reverting less useful fine-tuning updates after training often outperforms freezing them from the start.

1 Introduction

Post-training large language models (LLMs) on labeled text data is a critical step in developing models for real-world applications. However, when these LLMs are fine-tuned for lower-resource languages, significant challenges arise due to the pretrained model's limited capabilities. Although in recent years the broader scaling of pretraining and increased investment in additional languages (Dang et al., 2024b; Llama et al., 2024) have led to major improvements, pretrained LLMs still struggle to understand and generate text in all but a few languages (Romanou et al., 2025; Qin et al., 2025).

This pretraining disparity is further exacerbated by the lack of available high-quality multilingual fine-tuning data (Singh et al., 2024) and the significant cost to procure such annotated data (even through machine translation). For many of the capabilities developers target during post-training (e.g., instruction-following, reasoning, or safety) there are only sufficient open-source data available in English, Chinese, and a handful of other languages. This motivates the need for better cross-lingual transfer: the generalization of learned capacities from high-resource languages to lower-resource ones (Hu et al., 2020; Philippy et al., 2023).

Despite recent releases of massive mixture-of-expert LLMs (Team, 2024b; DeepSeek-AI et al., 2025; Team, 2025), a large majority of modern LLMs are *dense*, meaning that all parameters are active during training and inference. However, even within dense LLMs, recent works have found separability in where and how varying capabilities are represented (Yin et al., 2024; Yao et al., 2024). For example, multilingual capabilities are typically concentrated in the top and bottom transformer layers and multi-head attention parameters of an LLM (Chang et al., 2022; Choenni et al., 2024). This notably contrasts mathematical reasoning capabilities being encoded mainly in the middle transformer layers (Hanna et al., 2023; Stolfo et al., 2023). In the context of cross-lingual transfer, this functional separation motivates *modular* approaches to fine-tuning, which distinct model components can be trained, swapped, or merged (Bengio et al., 2020; Pfeiffer et al., 2023) for efficient and flexible multi-objective optimization.

In this work, we explore several modular approaches for composing target task and target language capabilities in off-the-shelf dense LLMs.

*Correspondence: lucasbandarkar@cs.ucla.edu

Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025), pages 131–148
November 8-9, 2025 ©2025 Association for Computational Linguistics

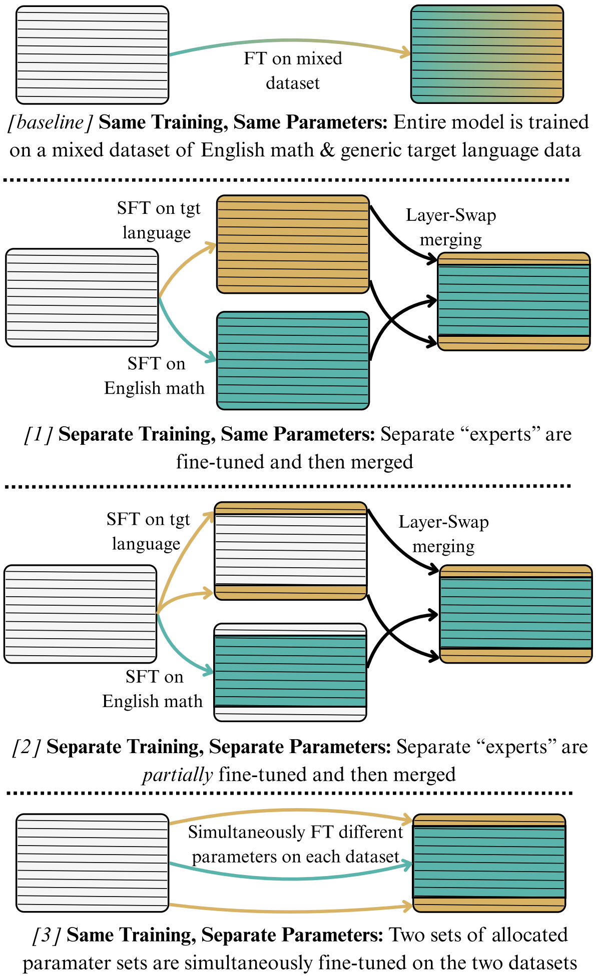

Figure 1: Illustration of the three methods that induce modularity by imposing target language capabilities (brown) and mathematical reasoning (blue) on separate LLM parameters. *[1]* is from Bandarkar et al. (2025a)

Our goal is to induce modularity by exploiting the differences in parameters that are most relevant to mathematical reasoning versus multilingual capabilities. We focus on the prevalent scenario where task-specific data is scarce in the target language but readily available in English. We address this by working with two datasets; one English math dataset for supervised fine-tuning (SFT) and one general, multi-task SFT dataset in the target language. Using the target languages of Bengali, Swahili, and Telugu, we evaluate the methods on the multilingual math benchmark, MGSM (Shi et al., 2023).

With these datasets, we evaluate numerous training paradigms that incentivize the model, to varying degrees, to learn multilingual or math capabilities in specific parameters. We organize the settings along two axes: (1) whether the models are optimized separately or together over the two SFT datasets and (2) whether the same or separate model parameters are trained on the datasets. When the models are trained separately, we combine the learned capabilities using model merging methods such as variants of Layer-Swapping (Bandarkar et al., 2025a). To train separate model parameters, we start by dividing all parameters into two partitions according to prior work: (1) one set allocated to target language training and (2) one set to English math. Only allocated parameters are fine-tuned, while the opposite partition is frozen. We additionally develop a method to train separate parameters in a single, joint training by frequently freezing and unfreezing parameters to simulate simultaneous training.

Despite the strong starting capabilities of the four LLMs and the data-constrained setting, our experimental results show that all of the modular solutions outperform our baselines, despite being subject to varying training constraints. This implies that intentional separation of parameters and/or training improves the *compositionality* of task and language capabilities.

Amongst our modular solutions, we surprisingly find that post hoc model merging via Layer-Swapping outperforms more coordinated multi-task fine-tuning approaches. To contextualize this counterintuitive result, we explore recent academic literature that help explain the phenomenon. We provide empirical evidence for training all model parameters, even if large portions will be discarded during Layer-Swapping. While these subsets of task vectors are unproductive, freezing them during fine-tuning leads to less optimal updates to the target parameters. Notably, we rationalize that the fine-tuning task vectors (Δs) are quite linear within individual parameter blocks (Dai et al., 2025), meaning they can be added, scaled, or interpolated as linear components (Adilova et al., 2024).

Overall, we enumerate the following principal contributions of this work:

- We develop and synthesize a number of modular solutions that *each* increase compositionality for cross-lingual transfer compared to non-modular baselines, demonstrated through extensive experiments.
- Of the modular methods, we find that fine-tuning all parameters and then merging via Layer-Swapping performs best on average.
- We provide a mix of theoretical and empirical explanations to explain the surprising success of Layer-Swapping relative to alternatives.

2 Background

2.1 Cross-Lingual Transfer

The relative abundance of textual data available in English in comparison to other languages has long motivated research in developing methods to efficiently transfer learned capabilities across languages (Koehn and Knight, 2002). Typically, some capabilities transfer naturally across languages, as evidenced by the superior performance of multilingual models on low-resource languages compared to monolingual models (Firat et al., 2016; Pires et al., 2019; Artetxe et al., 2020). In encoder models, the text embedding could be aligned across languages to improve transfer using methods such as contrastive learning (Mikolov et al., 2013; Artetxe et al., 2018; Muller et al., 2021).

However, cross-lingual alignment in more modern decoder-only models has become less methodical because of the lack of universal embedding (Kargaran et al., 2025). Since most popular LLMs have been trained on a majority English corpora, recent works have examined how much intrinsic cross-lingual transfer occurs at different training stages (Choenni et al., 2023; Wang et al., 2024). These large models have broader generalization and robustness, but still fail to transfer much of their capabilities across languages (Philippy et al., 2023). Recent works have identified prompting methods (Shi et al., 2023; Zhang et al., 2024) or post-training data augmentation (Dang et al., 2024a; She et al., 2024; Lai et al., 2024) to help generalization.

2.2 Modularity in Multilingual NLP

A major constraint for models being able to process many languages has been the number of parameters available to represent them. As a result, improving a language model in one language risks undermining its knowledge of another, termed the *curse of multilinguality* (Conneau et al., 2020; Pfeiffer et al., 2022). Naturally, numerous methods have been proposed to increase the model's parametric capacity without increasing the inference cost, such as mixture-of-expert architectures (Fedus et al., 2022) that route tokens according to their language (NLLB et al., 2022). Methods that leverage modular parameters were developed to compose capabilities for transfer learning by inserting trainable adapters within model layers (Houlsby et al., 2019; Pfeiffer et al., 2021). These methods were modified for multilinguality by allocating adapters for particular languages and switching them in or out depending on the input (Bapna and Firat, 2019; Pfeiffer et al., 2020). Pfeiffer et al. (2022) extended these methods by pretraining an adapter-based multilingual model from scratch. In decoder models, cross-lingual adapters have also been proposed at the token embedding level (Jiang et al., 2025).

Even in dense LLMs, however, interpretability research has identified the emergence of effective modularity (Csordás et al., 2021) as LLM parameters scale (Zhang et al., 2022; Qiu et al., 2024; Chen et al., 2025). Principally, numerous recent works have identified that just a few transformer layers at the top and bottom of English-centric LLMs are responsible for multilingual capabilities, notably by mapping input and output into a universal representation (Kojima et al., 2024; Wendler et al., 2024; Tang et al., 2024b; Alabi et al., 2024; Wu et al., 2025). Similar patterns are observed in modern sparse mixture-of-experts LLMs, where it is also observed that language-specialized experts are completely distinct from task/domain-specialized ones (Bandarkar et al., 2025b).

2.3 Model Merging

Model merging is the practice of combining the weights of multiple checkpoints of the same model architecture into a singular model. While averaging models is a fundamental machine learning approach to increase statistical robustness (Breiman, 1996), the averaging of model checkpoints, dubbed a model soup by Wortsman et al. (2022), has re-emerged in large-scale LLMs as a method to increase model robustness. More importantly, it also increases the search space of valid model variants at any given training step without additional costly training runs (Llama et al., 2024). However, simple weight averaging is vulnerable to negative transfer, or interference, between checkpoints so numerous methods have been presented to selectively merge parameters (Ilharco et al., 2023a; Yadav et al., 2023; Yu et al., 2024a). Surprisingly, training models on separate data and then merging can often outperform a single training run on mixed data (Tang et al., 2024a; Aakanksha et al., 2024) and has shown to be highly effective in large-scale multilingual pretraining (Dang et al., 2024b). For cross-lingual transfer in particular, Ansell et al. (2022) showed that sparse fine-tuning can lead to better composition. Bandarkar et al. (2025a) extended this by notably identifying that mathematical reasoning was concentrated in parameters different from multilingual capabilities. As a result, model

Training Description	Base Model	Partial LoRA	Partial SFT	LoRA	Full SFT
Math-only	19.0%	18.0%	19.5%	18.9%	19.6%
Language-only	19.0%	19.2%	19.8%	19.7%	20.3%
Data mixing	19.0%	-	-	19.7%	20.4%
Simultaneous SFT	19.0%	20.4%	21.0%	-	-
Layer-Swapping	19.0%	20.0%	20.4%	20.8%	**21.5%**

Table 1: Summary Table of Results. Each value represents *the average across four models, three languages, and multiple training runs* on MGSM in 2-shot evaluations. The last row represents "Separate Training" while the "Partial" trainings correspond to "Separate Parameters" trainings. All results shown here and in all other tables of this paper display exact-match (EM) accuracy (↑) as a percentage.

variants trained on English math data and multilingual data can be combined by Layer-Swapping, or swapping the transformer layers most important to each.

3 Experimental Setup

3.1 Evaluation

Limited by the lack of task-specific benchmarks for medium- and low-resource languages, we focus on MGSM (Shi et al., 2023) as the target task of this project. MGSM is a mathematical reasoning benchmark parallel across 10 languages as a result of high quality translations from the popular English benchmark, GSM8K (Cobbe et al., 2021). For MGSM, we report exact match accuracy in two-shot, as one- and zero-shot led to inconsistent results. More few-shot examples did not display substantial gain. For target languages, we choose the languages in MGSM where the four LLMs perform the worst: Bengali, Telugu, and Swahili. In addition, the lack of open-source math SFT data available in these languages motivates the need for more effective cross-lingual transfer. For a given fine-tuned model, we also evaluate the two-shot MGSM performance in English to evaluate its math performance irrespective of target language capability. Conversely, we use the multilingual MCQA benchmarks GLOBAL MMLU (Singh et al., 2025) and BELEBELE (Bandarkar et al., 2024) as pure language understanding signals, independent of math.

3.2 Models

We run experiments on four state-of-the-art instruction-finetuned LLMs: FALCON 3 7B (Team, 2024a), QWEN2.5 7B Instruct (Yang et al., 2024), LLAMA 3.1 8B Instruct (Llama et al., 2024), and AYA Expanse 8B (Dang et al., 2024b). All have similarly high performance on MGSM in English. LLAMA 3.1 and FALCON 3 are English-centric, QWEN2.5 bilingual with Chinese, and AYA Expanse explicitly multilingual. However, all officially cover numerous other languages (up to 23 for AYA) and perform reasonably on such languages, which we verify using BELEBELE and GLOBAL MMLU. Bengali, Swahili, and Telugu are amongst the official languages for none of these models. As a result, the four models are all low-scoring in MGSM in these languages, with the exception of LLAMA on Swahili (See Appendix A.8).

3.3 Parameter Allocation

To determine which parameters to "allocate" to each capability, we rely on a mix of interpretability papers and small-scale empirical tests. As mentioned in Section 2.2, numerous papers have identified the most important parameters for multilingual capabilities to be the first few and last few transformer layers of LLMs. These works, however, typically discuss mostly English-centric models (such as LLAMA 3.1 and FALCON 3). We therefore need to evaluate this for bilingual and multilingual models like QWEN2.5 and AYA Expanse. For mathematical reasoning, we note that Bandarkar et al. (2025a) identifies the middle and late-middle transformer layers as being the most important. This work, and numerous others (Voita et al., 2019; Ma et al., 2021; Zhao et al., 2024), similarly identifies multi-head attention parameters as critical to multilingual capabilities, as opposed to multi-layer perceptron parameters.

To empirically verify these assumptions on our selected models, we run SFT over our datasets with different subsets frozen. We evaluated numerous ways to partition the parameters and find a number of splits that enable improvements on English math and on language-specific signals (e.g. BELEBELE). To validate that the good performance when freez-

Parameters that are frozen or reset	Frozen during	Reset after
base (no SFT)	78.4%	78.4%
[Z] only top-4 and bottom-8 layers (inverse of intuition)	78.2%	78.9%
[A] all MHA parameters + MLP parameters in top-2 and bottom-6 layers	79.4%	79.8%
[B] only top-4 and bottom-8 layers	79.8%	79.8%
[C] only top-2 and bottom-6 layers	79.7%	80.0%
None	80.1%	80.1%

Table 2: MGSM 2-shot results (↑) on the *English* split after SFT on the English math data averaged across four models. These results (1) validate that our intuition leading to our parameter allocations *[A, B, C]* is reasonable seeing as results are close to full fine-tuning and are significantly higher than the inverse allocation *[Z]*. Additionally, (2) these results demonstrate that full fine-tuning then reverting parameters (second column) is more effective than freezing those parameters from the start (first column).

ing parameters is because the trainable parameters are particularly useful for a target task, we also run experiments with the *opposite* allocation (e.g. middle layers frozen during mathematical reasoning training) and find that it works poorly.

While the search space of which parameters to freeze is large, we settle on three partitions that show sufficient empirical success:

- *[A]* All multi-head attention parameters allocated to the target language. Then, amongst the multi-layer perceptron parameters, those in the first six and last two transformer layers still allocated to language, while those in the rest of the 32- or 36-layer LLM for math.
- *[B]* The first eight and last four transformer layers allocated to language, the rest for math.
- *[C]* The first six and last two transformer layers allocated to language, the rest for math.

In these three settings, both mathematical reasoning and target language capabilities improve similarly to full SFT with a fraction of trainable parameters (See Table 2 for results for math). We evaluate the three for each of our experimental settings and, unless noted, report the highest scoring.

3.4 Training

For SFT data, we create four datasets, one for math in English and one instruction dataset for each of the three target languages. The math instruction dataset consists of English math word problems from the Orca-Math synthetic dataset (Mitra et al., 2024). For the language datasets, we replicate the creation of "generic" instruction fine-tuning datasets from Bandarkar et al. (2025a) by combining samples from open-source instruction and task-specific datasets. Importantly, there are no math samples in these multi-task language datasets. We provide specific details and citations for these data collections in Appendix A.6.

Due to constraints on the amount of verifiable-quality data available in each of the target languages, our datasets are controlled at 80k samples, 2k of which is reserved for validation. Because of significantly diminishing returns exhibited by the validation loss and downstream evaluations, we only train for one epoch for each of our settings.

We additionally duplicate all experiments using Low-Rank Adapters (LoRA) (Hu et al., 2022). Specifically, we use rank-stabilized LoRA (Kalajdzievski, 2023) applied to both multi-layer perceptron and multi-head attention parameters. In general, the adjustments of our methods to be compatible with LoRA were minor unless noted otherwise. With four models, three languages, and two fine-tuning approaches (full and LoRA), we have a total of 24 experimental settings. For each, we do hyperparameter search over several runs to ensure comparability (See Appendix A.4 for details).

4 Experiments

We describe numerous methods that modularize off-the-shelf, dense LLMs in different ways. We describe *separate training* as when we conduct separate SFT runs on different datasets, albeit starting from the same off-the-shelf model. As previously mentioned, the separately trained checkpoints are then merged via Layer-Swapping. *Separate parameters* implies that only the partition of parameters *allocated* (See Section 3.3) to that dataset are trained while the rest remain frozen.

4.1 Baselines (Math-only and Language-only)

For comparison, we evaluate a number of straightforward SFT setups to serve as baselines. We do full-parameter training runs for each of the target language generic SFT datasets and the English math SFT dataset. For further baselines, we rerun the above when leaving only parameters *allocated* to that capability trainable, and the rest are frozen. In addition, we replicate both full training and partial training in LoRA, where parameters are "frozen" if no adapter is added for that parameter.

4.2 Data Mixing (Same Training, Same Parameters)

As an additional baseline, we randomly mix the two datasets together and jointly optimize over the two disjoint tasks with all parameters left trainable.

4.3 Layer-Swapping (Separate Training, Same Parameters)

For this setting, we exactly recreate the method presented by Bandarkar et al. (2025a). Starting from the same base model, separate variants are trained on different tasks, dubbed "experts". Concretely, one expert has been trained on the English math data, and the other on the target language instruction dataset. To recompose a single model, the top and bottom transformer layers from the target language expert replace those in the math expert, while the math experts' middle layers remain. We additionally implement the equivalent of this methodology with LoRA, where the set of adapters is merged by combining the adapters corresponding to parameters that would be swapped. Note that we do not retrain these experts and simply use the checkpoints from our baseline trainings.

4.4 Layer-Swapping with Partial SFT (Separate Training, Separate Parameters)

We modify Layer-Swapping so that only the parameters involved in the model merging are trained, and all those eventually ignored are kept frozen during training. The idea for this is that no parameters are unnecessarily trained and we can incentivize the training to focus the learned capabilities into the desired parameters. Similar to above, we do not retrain experts and simply merge checkpoints from our frozen parameter baselines.

4.5 Simultaneous Partition SFT (Same Training, Separate Parameters)

We design a methodology to "simultaneously" finetune two partitions of LLM parameters on two different datasets. To do so, we apply a gradient step on a batch from one dataset on the corresponding partition of parameters. Then, we switch which parameters are frozen and sample a batch from the other dataset for the next gradient step. This frequent back-and-forth is intended to ensure the coordination of parameter updates during multi-task optimization. The validation set contains an equal amount from each datasets.

Switching We default to a single step before switching to best simulate fully simultaneous training, but additionally experiment with more steps between. We set the effective batch size [1] to 64. At the end of each step, all parameters just updated are frozen for the next step and conversely, all frozen parameters are unfrozen. In addition, a flag for the data iterator is switched to ensure the next batch of data will be sampled from the appropriate dataset. For LoRA training, the same logic is implemented.

Optimizer We consider numerous approaches to adapt the AdamW optimizer (Loshchilov and Hutter, 2019) used in all previous experiments. Although we technically employ a single optimizer initialized on all parameters during training, we configure it to function as two independent optimizers, each exclusively managing its own separate subset of parameters. Namely, when a subset of parameters A is frozen, the corresponding AdamW optimizer states Ω_A (momentum and variance estimates) are also frozen in time. As a result, when the parameters in A are unfrozen, the corresponding momentum and variance estimates of Ω_A still reflect only the gradients steps previously applied to A. However, the other parameters A^c have been updated in the meantime, meaning Ω_A risks being outdated given the modified loss landscape. To test the impact of this inconsistency, we ablate over different numbers of steps between switches and find that the differences are very negligible (See Appendix A.3). We conclude that the optimizer restarting on an outdated loss landscape is of minimal concern, presumably because of the smoothness of the loss topology. Since there is a single optimizer, the learning rate schedule is the same for all (constant with warmup). And while the gradients tend to be larger for the multilingual data, we set a maximum gradient norm of 1.0 for clipping.

5 Results

Our experimental setting was designed to replicate a real-world scenario where multilingual LLM developers would take a post-trained LLM and are

[1]Effective batch size is the product of the batch size per GPU, number of GPUs, and gradient accumulation steps.

Performance Comparison of Modular Solutions

SFT Type	Base	Full	Simultaneous SFT		Layer-Swapping			
			Full	LoRA	Full SFT	LoRA	Part. SFT	Part. LoRA
Swahili	23.5%	25.1%	25.9%	25.2%	**26.7%**	25.8%	25.1%	24.8%
Bengali	25.6%	27.9%	27.9%	26.9%	**28.7%**	27.5%	27.0%	26.7%
Telugu	7.9%	8.2%	**9.3%**	9.0%	9.2%	9.2%	9.0%	8.6%
English	78.4%	80.4%	81.8%	80.5%	80.9%	80.8%	79.9%	80.0%
sw,bn,te AVG	19.0%	20.4%	21.0%	20.4%	**21.5%**	20.8%	20.4%	20.0%

Table 3: All values presented above are MGSM 2-shot EM accuracy (↑), averaged across four models. The baseline presented for comparison in the 3rd column is the full SFT on the mix of the two datasets.

limited by the amount of in-language post-training data. This constrained scenario means only modest improvements are achievable. However, we do observe several conclusive patterns. Across our different four models and three languages (12 *conditions*), we can summarize into 6 *treatments* discussed in Sections 4.1 to 4.5. Despite the small magnitude of differences, the rank-based Friedman test (non-parameteric) shows statistically significant differences between the *treatments* at the 0.05 significance level.

In our setting, we find that only training on the language dataset is more effective in improving the target language MGSM score than only on the math dataset (details in Appendix A.1). This implies, perhaps, that what our four models need most, is improved Swahili, Bengali, or Telugu abilities as opposed to math improvement.

We validate the lack of need for full-parameter training when doing both language adaptation and math SFT. Once the most useful parameters have been identified for such a skill, as discussed in Section 3.3, comparable performance to full SFT can be achieved with a fraction of the trainable parameters. Beyond potentially contributing to compositionality, this leads to faster and more memory-efficient training. More details on these baselines can be seen in Appendix A.1. We do note, however, that in the absence of resource limitations, SFT with less trainable parameters converged a bit slower and full fine-tuning still performed best. This is also true for LoRA, which has much less trainable parameters by nature.

A significant result is that all our modular solutions perform statistically-significantly better than the non-modular baselines, as can be seen in Table 1. This is strongly the case for Telugu and Swahili in the displayed four-model averages, but varies more by specific modular method for Bengali in comparison to the top baseline (data mixing) (See Appendix A.5 for per-language results).

Within our modular solutions, however, we find numerous surprising results. First, freezing the unused parameters in training experts before Layer-Swapping does not improve upon full training. As detailed in the last four columns of Table 3, the difference in performance is better when all modules are being finetuned for both LoRA and full-parameter SFT (statistically significant). This is counter-intuitive because the layers eventually merged are potentially dependent on parameter changes that are being replaced. Second, Layer-Swapping surprisingly outperforms the simultaneous SFT. This is surprising because in our simultaneous SFT, the modularity is being imposed cohesively as opposed to the ad hoc merging of layers from separate training runs. We note, however, that the simultaneous SFT performs second-best.

To validate results further, we also evaluate more expensive Continual Pretraining (CPT) for QWEN2.5 in Bengali across the experimental designs and find agreement with our SFT results (See details in Appendix A.2, A.7). However, we limit discussion of these results because of the small scale of experimental results.

We additionally analyze the composability of individual experts under Layer-Swapping. We define a good merging indicator as an evaluation signal *of an expert* that correlates with the performance of the merged model. We find that performance on general NLU benchmarks—BELEBELE and GLOBAL MMLU—is a stronger indicator of a *language* expert's merge quality than MGSM results in the target language. Similarly, MGSM performance in English is a better predictor for a *math* expert than MGSM in the target language. This is notable because MGSM in the target language is the target task of course, yet results more directly related to the training data tends to be more important for proper task composition.

6 Discussion

Given the rejection of our hypothesis that simultaneous fine-tuning would most effectively compose task and language capabilities, we discuss potential explanations for this outcome.

Train-then-Revert vs. Freeze-then-Train Intuition may dictate that fine-tuning parameters and then later reverting part of them should be less effective than simply freezing those parameters from the start. In the former, the fine-tuning is unaware of future edits while the latter provides hard constraints during optimization. However, empirically, we find that across models, training-then-resetting outperforms freezing-then-resetting. We display this for our English math fine-tuning in Table 3.3. This explains why Layer-Swapping with full training (Section 4.3) may be preferential to solutions involving freezing parameters. We conclude that while a large portion of fine-tuning weight updates are not needed in the end, either because they are noisy or redundant (Yu et al., 2024b), they enable optimization in a very high-dimensional space. This is analogous to recent papers discussing the Lottery Ticket Hypothesis (Frankle and Carbin, 2019), where it has been concluded that training a full neural network and then pruning it leads to stronger models than the same pruning before training (Frankle et al., 2021).

Concatenating Components in Layer-Swapping We seek to explain why concatenating transformer layers from separately fine-tuned "experts" is so seamless. Task vectors (Ilharco et al., 2023b) are the Δs that result from fine-tuning (i.e., $\theta_{FT} - \theta_0$). Task vector *linearity* refers to the property that linear combinations of such task vectors form a coherent, effective model. Ortiz-Jimenez et al. (2023) identifies that linearized task vectors exhibit better mergeability. Meanwhile, when fine-tuning heavily post-trained models like those used in our experiments, recent works show that updates to individual model layers exhibit significant linearity (Zhou et al., 2024; Razzhigaev et al., 2024; Dai et al., 2025). Furthermore, research on mode connectivity (Frankle et al., 2020; Garipov et al., 2018) shows individual transformer layers can be smoothly interpolated (Zhou et al., 2023; Adilova et al., 2024). These works provide explanation for why ad hoc Layer-Swapping is not more degradative.

Further Considerations We note that model merging is convenient because the configuration (e.g., what parameters to swap), can be determined after training. This enables fast iteration through configurations without retraining. This flexibility is sacrificed for our "separate parameters" methods, which require fixing parameter allocations. However, an inconvenience of merging methods is the need to train two experts, potentially doubling the amount of training runs for hyperparameter search.

7 Conclusions

Our results demonstrate that imposing modularity into dense LLMs for cross-lingual transfer is quite effective in low-data scenarios. We empirically validate this with numerous ways to impose such modularity through fine-tuning with frozen parameters or model merging, all of which prove more effective than non-modular baselines. Furthermore, we discover the surprising success of Layer-Swapping over other modular methods that fine-tune task and language together or do not ad hoc revert parameter updates. We conjecture that the success of this ad hoc merging method is because the math and language experts, when represented as task vectors, exhibit a high degree of linearity. As a result, this method benefits from more robust training over all parameters while also leading to effective compositionality. We also empirically demonstrate that the success of Layer-Swapping is in part due to frozen-parameter fine-tunings underperforming full fine-tunings followed by parameter resets.

8 Future Work

We encourage further work in multilingual NLP that leverages implicit modularity in LLMs, induces it during training, or designs explicitly modular architectures. Our parameter allocation strategy relied on previous interpretability work and limited empirical evidence, and the search space of modular configurations is largely unexplored. With post hoc model merging, iterating through many ablations can be quick. Although we focused on mathematical reasoning—due to limited multilingual task-specific datasets—future work should examine other tasks that may warrant different parameter allocations. More broadly, these results underscore the importance of improving interpretability around how capabilities are parameterized in LLMs, such as multilinguality. If we can better localize and separate parameters by function, our findings suggest that modularization may yield significant improvements.

Limitations

Small Δs Our decision to use the instruction fine-tuned version of each of the open-source LLMs for our experiments was a conscious one that came with many considerations. We prioritized replicating a real-life practical scneario, where model developers would start from already fine-tuned LLM versions because of their broader capabilities. However, as a result, this meant that our fine-tuning experiments only led to relatively small performance improvements with respect to the starting checkpoint. Such checkpoints have undergone extensive post-training, notably with significant mathematical reasoning samples and varying amounts of multilingual samples. Therefore, possible model improvements with these small datasets were small, risking results that were not statistically significant. Nevertheless, this allowed us to control for the amount of improvement on benchmarks that was simply a result of the LLMs' improved ability to follow instructions after SFT, in addition to reflecting a more practical setting.

Acknowledgement

The authors acknowledge the support provided by Tanmay Parekh and Mohsen Fayyaz for this project.

References

Aakanksha, Arash Ahmadian, Seraphina Goldfarb-Tarrant, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. 2024. Mix data or merge models? optimizing for performance and safety in multilingual contexts. In *Neurips Safe Generative AI Workshop 2024*.

Linara Adilova, Maksym Andriushchenko, Michael Kamp, Asja Fischer, and Martin Jaggi. 2024. Layer-wise linear mode connectivity. In *The Twelfth International Conference on Learning Representations*.

Jesujoba Alabi, Marius Mosbach, Matan Eyal, Dietrich Klakow, and Mor Geva. 2024. The hidden space of transformer language adapters. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6607, Bangkok, Thailand. Association for Computational Linguistics.

Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, Dublin, Ireland. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.

Lucas Bandarkar, Benjamin Muller, Pritish Yuvraj, Rui Hou, Nayan Singhal, Hongjiang Lv, and Bing Liu. 2025a. Layer swapping for zero-shot cross-lingual transfer in large language models. In *The Thirteenth International Conference on Learning Representations*.

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. 2025b. Multilingual routing in mixture-of-experts. *Preprint*, arXiv:2510.04694.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sebastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. 2020. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*.

Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

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*, pages 119–136, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuxin Chen, Yiran Zhao, Yang Zhang, An Zhang, Kenji Kawaguchi, Shafiq Joty, Junnan Li, Tat-Seng Chua, Michael Qizhe Shieh, and Wenxuan Zhang. 2025. The emergence of abstract thought in large language models beyond any language. *Preprint*, arXiv:2506.09890.

Rochelle Choenni, Dan Garrette, and Ekaterina Shutova. 2023. How do languages influence each other? studying cross-lingual data sharing during LM fine-tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13244–13257, Singapore. Association for Computational Linguistics.

Rochelle Choenni, Ekaterina Shutova, and Dan Garrette. 2024. Examining modularity in multilingual LMs via language-specialized subnetworks. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 287–301, Mexico City, Mexico. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2021. Are neural nets modular? inspecting functional modularity through differentiable weight masks. In *International Conference on Learning Representations*.

Rui Dai, Sile Hu, Xu Shen, Yonggang Zhang, Xinmei Tian, and Jieping Ye. 2025. Leveraging submodule linearity enhances task arithmetic performance in LLMs. In *The Thirteenth International Conference on Learning Representations*.

John Dang, Arash Ahmadian, Kelly Marchisio, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. 2024a. RLHF can speak many languages: Unlocking multilingual preference optimization for LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13134–13156, Miami, Florida, USA. Association for Computational Linguistics.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, and 26 others. 2024b. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *Preprint*, arXiv:2412.04261.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1).

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. 2020. Linear mode connectivity and the lottery ticket hypothesis. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3259–3269. PMLR.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. 2021. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry P Vetrov, and Andrew G Wilson. 2018. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

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*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023a. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023b. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Fan Jiang, Honglin Yu, Grace Chung, and Trevor Cohn. 2025. Franken-adapter: Cross-lingual adaptation of llms by embedding surgery. *Preprint*, arXiv:2502.08037.

Damjan Kalajdzievski. 2023. A rank stabilization scaling factor for fine-tuning with lora. *Preprint*, arXiv:2312.03732.

Amir Hossein Kargaran, Ali Modarressi, Nafiseh Nikeghbal, Jana Diesner, François Yvon, and Hinrich Schuetze. 2025. MEXA: Multilingual evaluation of English-centric LLMs via cross-lingual alignment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 27001–27023, Vienna, Austria. Association for Computational Linguistics.

Mohammed Safi Ur Rahman Khan, Priyam Mehta, Ananth Sankar, Umashankar Kumaravelan, Sumanth Doddapaneni, Suriyaprasaad B, Varun G, Sparsh Jain, Anoop Kunchukuttan, Pratyush Kumar, Raj Dabre, and Mitesh M. Khapra. 2024. IndicLLMSuite: A blueprint for creating pre-training and fine-tuning datasets for Indian languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15831–15879, Bangkok, Thailand. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pages 9–16, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. 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 6919–6971, Mexico City, Mexico. Association for Computational Linguistics.

Wen Lai, Mohsen Mesgar, and Alexander Fraser. 2024. LLMs beyond English: Scaling the multilingual capability of LLMs with cross-lingual feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8186–8213, Bangkok, Thailand. Association for Computational Linguistics.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gérard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, and 35 others. 2022. The bigscience ROOTS corpus: A 1.6TB composite multilingual dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Team Llama, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian andAhmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and et al. 2024. The llama 3 herd of models. *Meta Research*.

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

Weicheng Ma, Kai Zhang, Renze Lou, Lili Wang, and Soroush Vosoughi. 2021. Contributions of transformer attention heads in multi- and cross-lingual tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1956–1966, Online. Association for Computational Linguistics.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *Preprint*, arXiv:1309.4168.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. *Preprint*, arXiv:2402.14830.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2214–2231, Online. Association for Computational Linguistics.

Team NLLB, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Meta Research*.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2023. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, volume 36, pages 66727–66754. Curran Associates, Inc.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2024. Openwebmath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations*.

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, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Ponti. 2023. Modular deep learning. *Transactions on Machine Learning Research*. Survey Certification.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Fred Philippy, Siwen Guo, and Shohreh Haddadan. 2023. Towards a common understanding of contributing factors for cross-lingual transfer in multilingual language models: A review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5877–5891, Toronto, Canada. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S. Yu. 2025. A survey of multilingual large language models. *Patterns*, 6(1):101118.

Zihan Qiu, Zeyu Huang, and Jie Fu. 2024. Unlocking emergent modularity in large language models. 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 2638–2660, Mexico City, Mexico. Association for Computational Linguistics.

Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Nikolai Gerasimenko, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. 2024. Your transformer is secretly linear. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5376–5384, Bangkok, Thailand. Association for Computational Linguistics.

Angelika Romanou, Negar Foroutan, Anna Sotnikova, Sree Harsha Nelaturu, Shivalika Singh, Rishabh Maheshwary, Micol Altomare, Zeming Chen, Mohamed A. Haggag, Snegha A, Alfonso Amayuelas, Azril Hafizi Amirudin, Danylo Boiko, Michael Chang, Jenny Chim, Gal Cohen, Aditya Kumar Dalmia, Abraham Diress, Sharad Duwal, and 38 others. 2025. INCLUDE: Evaluating multilingual language understanding with regional knowledge. In *The Thirteenth International Conference on Learning Representations*.

Shuaijie She, Wei Zou, Shujian Huang, Wenhao Zhu, Xiang Liu, Xiang Geng, and Jiajun Chen. 2024.

MAPO: Advancing multilingual reasoning through multilingual-alignment-as-preference optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10015–10027, Bangkok, Thailand. Association for Computational Linguistics.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, and 4 others. 2025. Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergun, Ifeoma Okoh, and 14 others. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11521–11567, Bangkok, Thailand. Association for Computational Linguistics.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052, Singapore. Association for Computational Linguistics.

Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. 2024a. Merging multi-task models via weight-ensembling mixture of experts. In *Forty-first International Conference on Machine Learning*.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024b. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

Falcon-LLM Team. 2024a. The falcon 3 family of open models.

Qwen Team. 2025. Qwen3.

The Mosaic Research Team. 2024b. Introducing dbrx: A new state-of-the-art open llm. *Mosaic AI Research*.

Atnafu Lambebo Tonja, Bonaventure FP Dossou, Jessica Ojo, Jenalea Rajab, Fadel Thior, Eric Peter Wairagala, Aremu Anuoluwapo, Pelonomi Moiloa, Jade Abbott, Vukosi Marivate, and 1 others. 2024. Inkubalm: A small language model for low-resource african languages. *arXiv preprint arXiv:2408.17024*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Hetong Wang, Pasquale Minervini, and Edoardo Ponti. 2024. Probing the emergence of cross-lingual alignment during LLM training. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12159–12173, Bangkok, Thailand. Association for Computational Linguistics.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in English? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394, Bangkok, Thailand. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 23965–23998. PMLR.

Zhaofeng Wu, Xinyan Velocity Yu, Dani Yogatama, Jiasen Lu, and Yoon Kim. 2025. The semantic hub hypothesis: Language models share semantic representations across languages and modalities. In *The Thirteenth International Conference on Learning Representations*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. 2024. Knowledge circuits in pretrained transformers. In *Advances in Neural Information Processing Systems*, volume 37, pages 118571–118602. Curran Associates, Inc.

Fangcong Yin, Xi Ye, and Greg Durrett. 2024. Lofit: Localized fine-tuning on LLM representations. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning*. PMLR.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024b. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *ICML*.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. MoEfication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 877–890, Dublin, Ireland. Association for Computational Linguistics.

Zhihan Zhang, Dong-Ho Lee, Yuwei Fang, Wenhao Yu, Mengzhao Jia, Meng Jiang, and Francesco Barbieri. 2024. PLUG: Leveraging pivot language in cross-lingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7025–7046, Bangkok, Thailand. Association for Computational Linguistics.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhanpeng Zhou, Zijun Chen, Yilan Chen, Bo Zhang, and Junchi Yan. 2024. On the emergence of cross-task linearity in pretraining-finetuning paradigm. In *Forty-first International Conference on Machine Learning*.

Zhanpeng Zhou, Yongyi Yang, Xiaojiang Yang, Junchi Yan, and Wei Hu. 2023. Going beyond linear mode connectivity: The layerwise linear feature connectivity. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A Appendix

A.1 Detailed Baseline Results

Detailed Performance of Non-Modular Baselines									
SFT Dataset	**None**	**Data-Mixing**		**Math-Only**			**Language-Only**		
SFT Type	**Base**	**Full**	**LoRA**	**Full**	**LoRA**	**Part. FT**	**Full**	**LoRA**	**Part. FT**
Swahili	23.5%	25.1%	24.8%	**25.2%**	24.4%	25.0%	24.8%	23.8%	24.3%
Bengali	25.6%	27.9%	26.0%	26.1%	24.8%	25.6%	**28.3%**	26.6%	26.9%
Telugu	7.9%	8.2%	8.4%	7.4%	7.4%	8.0%	7.9%	**8.6%**	8.2%
English	78.4%	80.4%	80.0%	**81.3%**	81.0%	80.6%	79.9%	78.8%	79.0%
sw,bn,te AVG	19.0%	**20.4%**	19.7%	19.6%	18.9%	19.5%	20.3%	19.7%	19.8%

Table 4: All values presented above are MGSM 2-shot EM accuracy (↑), averaged across four models. Generally, we find that data mixing is the most effective, but with very small difference in comparison to language-only SFT. We exclude Partial LoRA results for space considerations, but report here that the results were for all numbers, 0-1% lower than LoRA results.

A.2 CPT Results for QWEN2.5 in Bengali

Detailed Performance of CPT Experiments									
SFT Dataset	**None**	**Mix**	**Math-Only**		**Lang-Only**		**Simult.**	**Layer-Swapping**	
SFT Type	**Base**	**Full**	**Full**	**Part.FT**	**Full**	**Part.FT**	**Part.FT**	**Full**	**Part.FT**
Bengali	37.6%	38.2%	33.2%	34.2%	37.6%	37.8%	38.8%	39.4%	38.8%
English	76.8%	77.6%	80.0%	79.8%	74.0%	73.8%	80.2%	79.2%	79.6%

Table 5: All values presented above are MGSM 2-shot EM accuracy (↑), averaged across two runs. We find that our main results from SFT mostly stand, but limit our conclusions as the small number of runs prevent the findings from being statistically significant. We note that CPT trainings more substantially degrade performance in the *opposite* capability than SFT. "Mix" is "Data-Mixing" and "Simult." is "Simultaneous FT", shortened for space.

A.3 Number of Gradient Steps Between Switches

Table 6: Ablation over the number of gradient steps to do on a single dataset and single partition of model parameters before switching back to the other data and parameters. All runs were controlled to have the same exact hyperparameter settings on QWEN2.5 7B Instruct with the target language Swahili. Four upper layers and eight lower layers were allocated for the target language, and a learning rate $1.2e^{-06}$

Gradient Steps per Switch	**Starting Validation Loss**	**Ending Validation Loss**	**Δ for MGSM, Swahili**
1	2.301	1.605	+3.2%
5	2.301	1.612	+2.4%
10	2.301	1.613	+2.8%
50	2.301	1.613	+2.0%
200	2.301	1.602	+0.8%
500	2.301	1.565	+1.2%
1171	2.301	1.536	−1.2%

These results indicate no negligible differences between the tested step counts. This implies the concern discussed in Section 4.5 of the optimizer unfreezing with an outdated loss landscape is minimal. Or at least, it implies that the ability to do numerous steps without interruption in the same setting outweighs this concern. And while increasing the gradient steps per switch does provide no negligible difference on the validation loss, intuitively it leads to a training paradigm farther from a truly simultaneous training. We find that on the target task, MGSM in Swahili, performance goes down progressively as the gradient steps per switch is increased. This implies the composition of math and Swahili capabilities are working less effectively.

A.4 Details for Reproducibility

For reproducibility, we detail our implementation and hyperparameters for training. The datasets themselves are outlined in Sections A.6 and A.7.

- Training is run on a single cluster of A100s, typically with only one GPU per training run.
- Training methods are developed using the trl python package (von Werra et al., 2020) and models accessed via HuggingFace.
- Learning rate ranged across training runs, but was typically in the range $[1.0, 2.0] \times 10^{-6}$.
- For LoRA, it ranged from $[4.0, 9.0] \times 10^{-6}$. Rank and Alpha parameters were either $(64, 16)$ or $(32, 8)$.
- Sequence length was either 512 or 1024. Effective batch size was typically 32, except for effective batch size of 64 for simultaneous training, as described in Section 4.5.
- Evaluation is performed using the Language Model Evaluation Harness (Gao et al., 2024).

A.5 Bar Graph of Per-Language Results

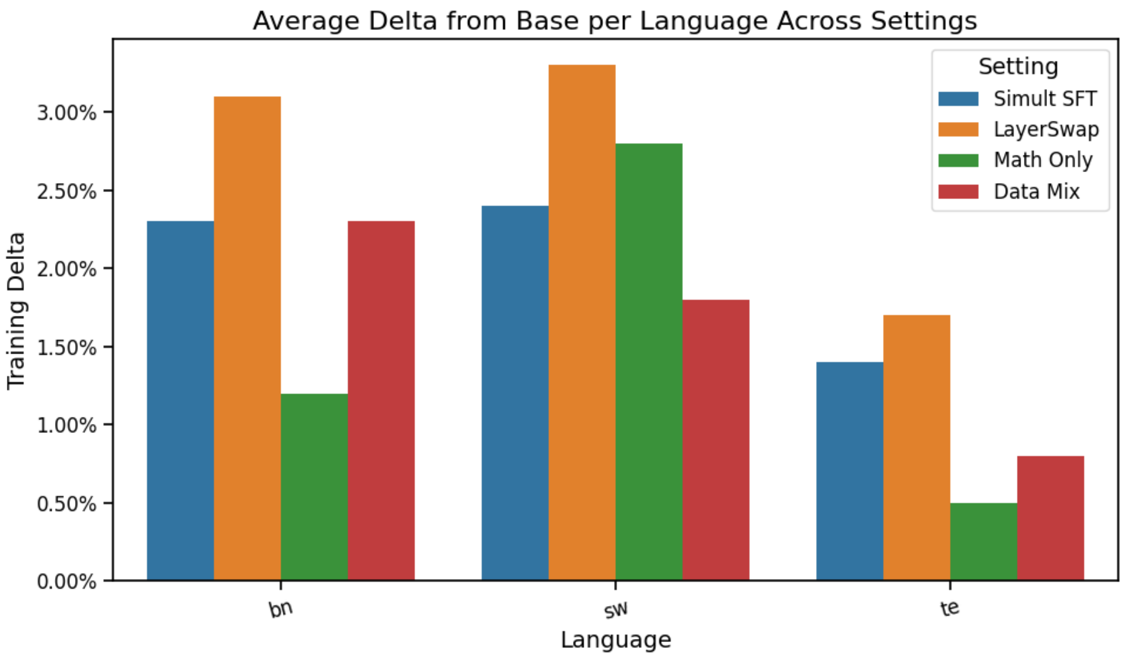

Figure 2: Per-language breakdown of the average performance gain seen during our different types of training, averaged across four models. We see that while math-only SFT (green) does well for Swahili and mixed-data SFT (red) does well for Bengali, our two modular solutions work consistently well across the three languages. Note: the y-axis is a percentage because the evaluation score is accuracy, *not* because this table displays percent change.

A.6 SFT Datasets

Table 7: Datasets used for supervised-fine-tuning (SFT) in this project

Category	Datasets	URL
Math	Orca Math word problems dataset from Microsoft (Mitra et al., 2024)	https://huggingface.co/datasets/microsoft/orca-math-word-problems-200k
Telugu	Aya Dataset from Cohere for AI (Singh et al., 2024)	https://huggingface.co/datasets/CohereForAI/aya_dataset
	NLLB English-Telugu translation data from FAIR (NLLB et al., 2022)	https://huggingface.co/datasets/allenai/nllb
	Synthetic English instruction dataset, machine translated to Telugu by Telugu-LLM-Labs	https://huggingface.co/collections/Telugu-LLM-Labs/indic-alpaca-datasets-65f2a3687d5cdbce8880c581
Bengali	Aya Dataset by Cohere for AI (Singh et al., 2024)	https://huggingface.co/datasets/CohereForAI/aya_dataset
	NLLB English-Bengali translation data from FAIR (NLLB et al., 2022)	https://huggingface.co/datasets/allenai/nllb
	IndicShareLlama dataset from AI4Bharat (Khan et al., 2024)	https://huggingface.co/datasets/ai4bharat/indic-align
	BongChat dataset from Lumatic AI	https://huggingface.co/datasets/lumatic-ai/BongChat-v1-253k
Swahili	Aya Dataset by Cohere for AI (Singh et al., 2024)	https://huggingface.co/datasets/CohereForAI/aya_dataset
	NLLB English-Swahili translation data from FAIR (NLLB et al., 2022)	https://huggingface.co/datasets/allenai/nllb
	Inkuba dataset from Lelapa (Tonja et al., 2024)	https://huggingface.co/datasets/lelapa/Inkuba-instruct
	xP3 MT dataset from BigScience, with FLoRes samples removed (Muennighoff et al., 2023)	https://huggingface.co/datasets/bigscience/xP3mt

All datasets listed above were verified to be used in compliance with their respective licenses. Each dataset was properly attributed according to its license requirements.

A.7 CPT Datasets

Table 8: Datasets used for continual pretraining (CPT) in this project

Category	Datasets	URL
Math	Open Web mathematical texts collected by the University of Toronto and Cambridge (Paster et al., 2024)	https://huggingface.co/datasets/open-web-math/open-web-math
Bengali	The ROOTS corpus subset of Bengali Wikipedia from BigScience (Laurençon et al., 2022)	https://huggingface.co/datasets/bigscience-data/roots_indic-bn_wikisource

All datasets listed above were verified to be used in compliance with their respective licenses. Each dataset was properly attributed according to its license requirements.

A.8 Off-the-shelf Model Results

To motivate the use of our four models and the three target languages, we provide preliminary results of these models prior to any fine-tuning.

Model	Size	MGSM				BELEBELE			
		EN	SW	BN	TE	EN	SW	BN	TE
LLAMA 3.1	8B	79.6%	52.0%	32.8%	11.2%	88.6%	56.1%	59.3%	53.6%
QWEN2.5	7B	76.8%	12.8%	37.6%	13.6%	91.1%	37.2%	64.7%	41.3%
AYA Expanse	8B	78.8%	10.8%	21.6%	3.2%	81.6%	32.3%	42.3%	29.9%
FALCON 3	7B	78.4%	14.4%	10.4%	3.6%	85.9%	36.3%	34.8%	30.1%

Table 9: The results on the MGSM (2-shot, EM accuracy (↑)) and BELEBELE (0-shot accuracy (↑)) benchmarks for the four models used in our experiments. We note that for all models, we use the instruction-finetuned version.