Mitigating Catastrophic Forgetting in Language Transfer via Model Merging

Anton Alexandrov[*1], **Veselin Raychev**[1,2], **Mark Niklas Müller**[2,3]
Ce Zhang[1,4,5], **Martin Vechev**[1,3], **Kristina Toutanova**[1,6]
[1] INSAIT, Sofia University "St. Kliment Ohridski" [2]LogicStar.ai
[3] ETH Zurich [4] University of Chicago [5] Together AI [6] Google DeepMind

Abstract

As open-weight large language models (LLMs) achieve ever more impressive performances across a wide range of tasks in English, practitioners aim to adapt these models to different languages. However, such language adaptation is often accompanied by catastrophic forgetting of the base model's capabilities, severely limiting the usefulness of the resulting model. We address this issue by proposing Branch-and-Merge (BAM), a new adaptation method based on iteratively merging multiple models, fine-tuned on a subset of the available training data. BAM is based on the insight that this yields lower magnitude but higher quality weight changes, reducing forgetting of the source domain while maintaining learning on the target domain. We demonstrate in an extensive empirical study on Bulgarian and German that BAM can significantly reduce forgetting while matching or even improving target domain performance compared to both standard continued pretraining and instruction finetuning across different model architectures.

1 Introduction

Large language models have shown remarkable capabilities, particularly in English. However, for less prevalent languages, performance can be significantly lower, making additional adaptation paramount (Zhao et al., 2024; Cui and Yao, 2024).

Catastrophic Forgetting Unfortunately, most adaptation techniques come at the cost of catastrophic forgetting of the base model's capabilities (Zhai et al., 2023; Shi et al., 2024; Li and Lee, 2024; Gogoulou et al., 2023). At the same time, retaining these capabilities is often crucial for solving downstream tasks in a new language. For example, math and coding skills learned in English can be extremely helpful for general problem-solving or reasoning tasks in other languages.

* Correspondence author: anton.alexandrov@insait.ai

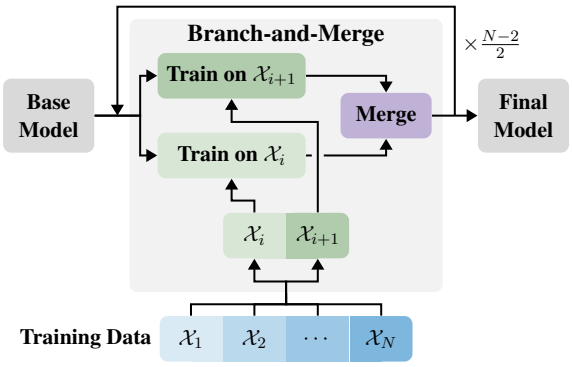

Figure 1: Illustration of Branch-and-Merge (BAM). We first split the training data into N slices (blue ■). We then iteratively finetune the current base model on two of these slices (green ■) and merge the resulting models to obtain the base model for the next iteration (purple ■). We repeat this until all N data slices have been used.

Experience Replay To mitigate such catastrophic forgetting, mixing in source language data in the target language training set, so-called experience replay, has proven effective for both continued pretraining (Ibrahim et al., 2024) and instruction tuning (Scialom et al., 2022; Zhang et al., 2023). However, experience replay alone can not fully mitigate forgetting. Especially when the exact source data is unknown (e.g. for state-of-the-art language models), experience replay can only be implemented approximately, reducing its effectiveness and necessitating further regularization.

This Work: Mitigating Catastrophic Forgetting with Branch-and-Merge We build on ideas from continual learning and introduce Branch-and-Merge (BAM – illustrated in Fig. 1), a novel method for adapting pretrained language models to new languages, underrepresented in their unknown training data, while minimizing the loss of previously learned capabilities. Concretely, BAM splits the training data into N slices (blue ■ in Fig. 1), before iteratively training the current base model on K (here two) such data slices in parallel (green

Findings of the Association for Computational Linguistics: EACL 2024, pages 17167–17186
November 12-16, 2024

■) and finally merging them (purple ■) to obtain the initial model for the next iteration. This significantly reduces the total weight change and as a result, forgetting, while preserving most of the learning from the parallel training steps. In particular, while target language perplexity is slightly increased compared to standard continued training, the retained base model skills lead to higher downstream performance on target language tasks.

Results We apply BAM to adapt MISTRAL-7B (Jiang et al., 2023) and LLAMA-3-8B (AI@Meta, 2024b) from predominantly English to an alphabet-sharing (German) and a non-alphabet-sharing (Bulgarian) language, considering both continued pretraining and instruction tuning.

We show that BAM consistently improves benchmark performance in both the target and source language compared to standard training, while not incurring additional computational or data costs. For example, when applied to instruction tuning, BAM significantly improves performance, allowing our LLAMA-3-8B BAM-trained for Bulgarian to outperform LLAMA-3-8B-Instruct not only in Bulgarian (by 10.9%) but also in English (by 1.3%) by inducing smaller magnitude but more efficient weight changes. In particular, we show that BAM induces more favorable trade-offs between learning and forgetting than prior techniques such as reduced learning rates (Winata et al., 2023) and LORA (Biderman et al., 2024).

Key Contributions Our main contributions are:

- We propose Branch-and-Merge (BAM), a training technique for language adaptation, improving learning while mitigating forgetting (Section 3).
- We develop a high-quality data mix for approximate experience replay significantly improving language transfer (Section 4).
- We conduct an extensive empirical investigation demonstrating the effectiveness of BAM across two target languages (Section 6).

2 Model Merging

A wide range of model merging methods have been proposed (Matena and Raffel, 2022; Yadav et al., 2023; Stoica et al., 2023; Yu et al., 2023; Wortsman et al., 2022). We experiment with LINEAR (Wortsman et al., 2022), SLERP (Goddard et al., 2024; Shoemake, 1985) and MODEL STOCK (Jang et al., 2024) merging, focusing on the first two, explained below. Let us consider the pretrained base model, f_θ, parameterized by θ which was finetuned on two different datasets $\mathcal{X}_1$ and $\mathcal{X}_2$, yielding f_{θ_1} and f_{θ_2}, respectively. We call the changes in weight due to this finetuning the task vectors $\tau_i = \theta_i - \theta$. To obtain a single model combining the learning from both datasets, we now merge these models.

LINEAR model merging interpolates task vectors or equivalently parameterizations linearly so to obtain $\theta' := \text{LINEAR}(\theta_1, \theta_2, c) = (1-c)\,\theta_1 + c\,\theta_2$.

SLERP first represents task vectors in polar coordinates before interpolating to obtain the new parameterization $\theta' = \tau' + \theta$

$$\vartheta = \arccos \frac{\tau_1 \cdot \tau_2}{|\tau_1| \cdot |\tau_2|}$$
$$\tau' = \frac{\sin((1-c)\,\vartheta)}{\sin(\vartheta)}\tau_1 + \frac{\sin(c\,\vartheta)}{\sin(\vartheta)}\tau_2$$

where ϑ is the angle between the two parameterizations and c is the interpolation coefficient. By slight abuse of notation, we write $\text{SLERP}(\theta_1, \theta_2, c)$ for both the resulting parameters θ' and the corresponding model $f_{\theta'}$.

3 Branch-and-Merge for Mitigating Forgetting in Language Transfer

To adapt a model f_θ pretrained on a typically unknown data distribution $\mathcal{X}_{\text{pre}}$ to a new task (language) without suffering from catastrophic forgetting, we propose the Branch-and-Merge (BAM) method, visualized in Fig. 1. BAM is based on first splitting the available training data into N slices (blue ■ in Fig. 1), and then iteratively training K models in parallel on one slice each (green ■) before merging the resulting models to obtain the base model for the next training iteration (purple ■). We first provide the intuition behind BAM before describing it in more detail.

Intuition There are two key ideas underlying BAM. First, lower magnitude weight changes τ_i, called task vectors, lead to less forgetting but also less learning. Second, the randomness in finetuning leads to task vectors $\tau_i = \tau^* + \epsilon_i$ with an unbiased error ϵ_i around the locally optimal task vector τ^* (Jang et al., 2024). We can thus reduce forgetting by reducing the task vector magnitude while offsetting the reduced learning by increasing task vector quality, i.e., reducing the error ϵ. If this error is unbiased and empirically Gaussian $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$

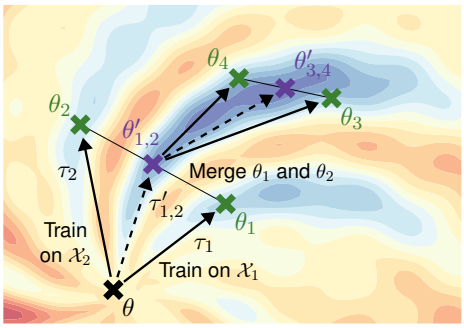

Figure 2: Illustration of BAM in the loss surface over parameter space. Both θ_1 and θ_2 land in poor local minima but their merge $\theta'_{1,2}$ lies in the valley of a better minimum. Training from there, θ_3 and θ_4 land at the boundary of that minimum due to noise in the training process and limited data. Their merge $\theta'_{3,4}$ cancels these errors and lies in the better minimum.

(Jang et al., 2024), merging, i.e., averaging, K noisy task vectors to obtain $\tau' = \frac{1}{K}\sum_{i=1}^{K}\tau_i$ reduces the corresponding expected error magnitude with $||\epsilon'||_2 \propto \frac{1}{\sqrt{K}}$, as $\tau' \sim \mathcal{N}(\tau^*, \frac{1}{K}\sigma^2)$. At the same time, increasing K in BAM reduces the number of consecutive training iterations (as more data slices are used per iteration) and thus the expected total weight magnitude which in turn reduces both learning and forgetting. This allows BAM to trade off learning and forgetting.

We visualize this in Fig. 2 for $K = 2$. There, the first two task vectors τ_1 and τ_2 land in the basins of poor local minima, with their merge $\theta'_{1,2}$ falling into the basin of a better minimum, highlighting the importance of BAM's iterative merging approach. Training $\theta'_{1,2}$ on two more data slices yields noisy task vectors τ_3 and τ_4 at the edge of this loss basin with their merge $\theta'_{3,4}$ falling right in the middle. In contrast, simply reducing the learning rate can also reduce task vector magnitude but does not improve task vector quality. We note that the same intuitions apply to SLERP and MODEL STOCK merging.

Implementation In more detail, we first partition the training data $\mathcal{X}_{\text{train}}$ into N not necessarily i.i.d. or equal-sized data slices $\mathcal{X}_i$. Then, we choose a parallelism factor K ($K = 2$ for most experiments and the visualizations in Figs. 1 and 2) and train our current base model f_θ independently on K of these data slices yielding f_{θ_i} to $f_{\theta_{i+K-1}}$. We merge the resulting models to obtain the base model for the next iteration $f_{\theta'} = \text{MERGE}(\{f_{\theta_j}\}_{j=i}^{i+K-1}, c)$. We typically choose the merging coefficient $c = 0.5$ but note that we can easily perform a 1-d line search over the resulting models. We then set the merged model $f_{\theta'}$ to be the base model $f_\theta \leftarrow f_{\theta'}$ for the next training iteration and repeat this process until we have used all data slices. We formalize this approach in Algorithm 1.

Algorithm 1 Branch-and-Merge (BAM)

Require: K: parallelism factor, f_θ: base model, $\{\mathcal{X}_i\}_{i=1}^{N}$: data slices, c: merging coefficient

$\Theta \leftarrow \{\}$
for $i \in [N]$ **do**
  $f_{\theta_i} \leftarrow \text{train}(f_\theta, \mathcal{X}_i)$
  $\Theta \leftarrow \Theta \cup \{\theta_i\}$
  if $i \mod K = 0 \,||\, i = N$ **then**
    $\theta \leftarrow \text{MERGE}(\Theta, c)$
    $\Theta \leftarrow \{\}$
return f_θ: finetuned model

4 Data Mixtures for Mitigating Forgetting in Language Transfer

Here we describe the data we use for continued pretraining of predominantly English base language models in order to adapt them to other languages. Outside of training methodology, we find in agreement with prior work that high-quality dataset mixtures are paramount for both effective language adaptation and reducing forgetting. We distinguish between experience replay of source language data and target language training data.

4.1 Approximate Experience Replay of Source Domain Data

While experience replay is crucial to alleviate forgetting (Rolnick et al., 2019; Ibrahim et al., 2024), the training data of most state-of-the-art models remains undisclosed. We, therefore, rely on approximate experience replay, constructing our approximate source data based on prior work (Penedo et al., 2023; Together.ai, 2023; Touvron et al., 2023; Groeneveld et al., 2024).

In more detail, we create a dataset consisting of OpenWebText (Gokaslan et al., 2019) - an open-source recreation of WebText (Radford et al., 2019), English Wikipedia, GitHub repositories, and a range of instruction finetuning datasets with a total of 15.1B unique tokens (see Table 1). We repeat the smaller IFT datasets 4 times to obtain an effective dataset size of 17.1B tokens. We note that while pretraining datasets commonly contain some instruction/response pairs, for example from Reddit, our experience replay mix most likely contains a higher portion of instruction data than the unknown source distribution.

Table 1: Composition of the approximate experience replay dataset. We report the number of unique tokens, how often a dataset is repeated (Rep.), and the resulting sampling probability (Prob.).

Dataset	Domain	#Tokens	Rep.	Prob. [%]
OpenWebText	Web	8.5B	1	49.8
Wikipedia-EN	Wiki	4.6B	1	26.9
GitHub repos	Code	1.35B	1	7.9
OpenHermes-2.5	IFT	357M	4	8.4
SlimOrca	IFT	197M	4	4.6
MetaMathQA	IFT	85M	4	2.0
CodeInstructions	IFT	20M	4	0.47

4.2 Minimal Experience Replay of Source Domain Data

To explore the significance of high-quality data for experience replay, we contrast the aforementioned approximate experience replay with what we call *minimal* experience replay. In minimal experience replay, we exclusively utilize samples from OpenWebText (Gokaslan et al., 2019), instead of a carefully curated data distribution. While minimal experience replay still incorporates source domain data during continuous pertaining, we anticipate it to cause a greater distribution shift than approximate experience replay. We chose the minimal experience replay to comprise roughly one-eighth of the training data (5B tokens for German and 10B for Bulgarian).

4.3 Constructing Target Language Data

While designing an optimal training data mix is still an open research problem (Xie et al., 2023; Tirumala et al., 2023; Shen et al., 2023), some key components have been identified that we adhere to for our target domain data. In particular, it has been shown that a small portion of code can notably improve the resulting reasoning capabilities and should thus be included (Liang et al., 2022; Ma et al., 2023; Fu and Khot, 2022). Furthermore, the importance of reasoning and instruction-following capabilities for end tasks suggests that instruction data would benefit the continued pretraining data mix. This agrees well with Jiang et al. (2024) suggesting a pre-instruction tuning phase to improve learning from new documents in continued pretraining. We discuss the exact data mixes we use in Section 5.2.

Bulgarian Training Data We adapt the RedPajama v2 pipeline (Together.ai, 2023) for Bulgarian/Cyrillic to annotate 84 Common Crawl * snapshots with a total of 30T tokens. After aggressive quality filtering and near-deduplication, we obtain a dataset of 50 to 80B Bulgarian tokens, depending on tokenization. We augment this dataset using the Bulgarian split of publicly available multilingual high-quality datasets such as Wikipedia, Eur-lex (Baisa et al., 2016), Europarl (Koehn, 2005), Parlamint (Erjavec et al., 2023), books, and a selection of private datasets containing news articles, legal texts, and literature. We further include selected machine-translated instruction data. See Table 2 for a full list of datasets. This yields a total of 77.7B unique tokens (using the original LLAMA-3 tokenizer) which we boost to 82.1B tokens by repeating the smaller and particularly high-quality datasets between 2 and 4 times.

Table 2: Composition of the Bulgarian target domain dataset. We report the number of unique tokens, how often a dataset is repeated (Rep.), and the resulting sampling probability (Prob.).

Dataset	Domain	#Tokens	Rep.	Prob. [%]
RPv2-BG	Web	70B	1	85.3
Legal docs	Legal	4.3B	1	5.2
Books	Literature	2.4B	2	5.9
Eur-Lex-BG	Legal	337M	2	0.82
Wikipedia-BG	Wiki	251M	4	1.2
OrcaMath-BG	IFT	100M	4	0.49
Bulgarian Law	Legal	58M	4	0.28
Parlamint-BG	Transcripts	52M	3	0.19
Curlicat	Mixed	40M	2	0.10
SlimOrca-BG	IFT	36M	4	0.18
CodeInstructions-BG	IFT	26M	4	0.13
Europarl-BG	Transcripts	24M	3	0.09
MetaMath-BG	IFT	15M	4	0.07
Open-Platypus-BG	IFT	13M	4	0.06

German Training Data German is significantly more abundant than Bulgarian in the quantity of text available from public datasets. We thus subsample roughly 10% of the German subset from the curated web text dataset CulturaX (Nguyen et al., 2024) equal to 41B LLAMA-3 tokens and include three German IFT datasets. For more details, see Table 15.

5 Experimental Setup

We now describe the experimental setup used to validate BAM's effectiveness for language adaptation. In particular, we discuss the target languages (Bulgarian and German), evaluation benchmarks (Section 5.1), training data (Section 5.2), and training setup (Section 5.3). We experiment with both continued pretraining of base models and instruction tuning of the resulting models.

*https://commoncrawl.org/

5.1 Target Languages and Benchmarks

To evaluate the effectiveness of BAM we conduct experiments on the transfer from general purpose, predominantly English models, to an alphabet-sharing (German) and a non-alphabet-sharing (Bulgarian) language, evaluating the resulting models on both the target and source languages.

While there is a large and growing number of high-quality datasets for evaluating LLMs in English and to a lesser extent German, these are much sparser for low-resource languages such as Bulgarian. We therefore first provide a brief overview of the English and German benchmarks we use before discussing the construction of a holistic evaluation suite for Bulgarian.

English Benchmarks We consider the following domains and benchmarks in English: *commonsense reasoning* (HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021), ARC-Easy, ARC-Challenge (Clark et al., 2018)), *multitask capabilities* (MMLU (Hendrycks et al., 2021)), *math* (GSM8K (Cobbe et al., 2021), MathQA (Amini et al., 2019)), and *reading comprehension* (Belebele English (Bandarkar et al., 2023), TriviaQA (Joshi et al., 2017)). We provide a detailed description of these benchmarks in Appendix B.1.

German Benchmarks We use the German benchmarks available in the Language Model Evaluation Harness (Gao et al., 2023). Some of these benchmarks are translated from English using GPT 3.5 (Plüster, 2023a) (TruthfulQA-DE, HellaSwag-DE, MMLU-DE, ARC-DE). We further consider human curated or translated benchmarks for *math* (MGSM-DE (Shi et al., 2023)), *paraphrasing* (PAWS-X (Yang et al., 2019)) and *reading comprehension* (BeleBele German (Bandarkar et al., 2023)). A detailed description of these benchmarks can be found in Appendix B.1.

Bulgarian Benchmarks As the number of publicly available Bulgarian benchmarks is limited, we translate all of the above English benchmarks using a combination of machine translation and over 600 hours of professional translators' work. We denote the translated benchmarks by appending '-BG' to their name and make them publicly available. In addition, we use the following Bulgarian benchmarks: *natural language inference* (XNLI (Conneau et al., 2018)) and *high-school exams* (EXAMS (Hardalov et al., 2020), MON-Tests). From these, XNLI was constructed through a professional translation of English examples by Conneau et al. and the other two are natively Bulgarian. We provide more details on the construction of the translation process and the novel MON-Tests benchmark in Appendix B.2.

Evaluation Metrics We aim to measure both learning, i.e., language adaptation, and forgetting. To this end, we consider benchmark scores and perplexity in the source and target language. Since our approximate experience replay data contains instruction tuning examples which can lead to improved English benchmark scores compared to the base model, we focus on held-out English document perplexity as a measure of forgetting. We use both benchmark performance (normalized accuracy) and held-out document perplexity as a measure of learning in the target language (see Appendix C for more details).

For both English and Bulgarian, we evaluate MMLU, TriviaQA, and EXAMS in a 5-shot, GSM8K in an 8-shot, and all other benchmarks in a zero-shot setting. All German benchmarks are run in a 5-shot setting.

5.2 Training Data

Below, we discuss the training data used for language adaptation.

Continued Pretraining Data For German, we subsample the training data including the approximate (17B tokens) and minimal experience replay (5B tokens) to 50B and 40B tokens respectively and divide it into $N = 4$ i.i.d. slices of 12.5B and 10B tokens each. For Bulgarian, we split the full 82B tokens of Bulgarian data plus 17B tokens of approximate or 10B tokens of minimal experience replay into $N = 8$ slices either i.i.d. or via a curriculum where the even-numbered slices contain significantly more experience replay data than the odd ones (see Table 17 in Appendix C).

Instruction Finetuning Data We investigate the effectiveness of BAM for instruction finetuning after continued pretraining. We collect $928K$ samples of English finetuning data and mix it with German or Bulgarian data. For Bulgarian, we generate $78K$ samples by using a mix of machine translation and professional translators to translate English samples to Bulgarian. For German, we use a mix of available, translated German IFT datasets. Please see Tables 13 to 15, as well as Appendix C for details on the resulting dataset.

5.3 Training Setup

Base models We chose MISTRAL-7B (Jiang et al., 2023) and LLAMA-3-8B (AI@Meta, 2024b) as base models due to their exceptional performance for their size and permissive licenses.

Details We implement BAM in PyTorch (Paszke et al., 2019) using HuggingFace's transformers library (Wolf et al., 2019) and DeepSpeed (Rasley et al., 2020; Rajbhandari et al., 2020a). We train each model on 64 NVIDIA H100s. Based on prior work and initial experiments, we find that 10^{-5} is the best maximum learning rate for continued pretraining on the models that we are using together with a batch size of 512 for continued pretraining and 256 for supervised finetuning. We use cosine decay to $0.1 \cdot$ max_lr with $\max(100, 0.01 \cdot \text{total\_steps})$ linear warmup.

6 Experimental Evaluation

We now evaluate BAM empirically for both continued pretraining (CPT) and instruction finetuning (IFT) before conducting extensive ablations and providing further results in Appendix A.

6.1 BAM for Continued Pretraining

Bulgarian CPT We use our data mix of Bulgarian data and English experience replay to adapt both LLAMA-3-8B and MISTRAL-7B to Bulgarian, comparing standard CPT and BAM in Table 3. We first demonstrate on MISTRAL that BAM with i.i.d. data slices matches standard CPT in Bulgarian (0.05% average score difference) while reducing forgetting significantly (20% less English NLL increase), achieving a 1.7% higher average score on English benchmarks and even outperforming the base model. Using our curriculum slices (only called BAM), we outperform standard CPT in 11 out of 12 Bulgarian benchmarks while retaining the reduced forgetting. Similarly, BAM achieves both a slightly higher average Bulgarian (0.3% better) and a notably higher English score (1.4% better) for LLAMA-3. We observed consistently across all of these settings that while standard CPT achieves a lower negative log-likelihood (NLL) in Bulgarian, indicating it fits the Bulgarian language modeling task better, the increased forgetting of base model capabilities (higher English NLL) leads to worse or equal benchmark performance.

German CPT We observe very similar trends adapting LLAMA-3-8B to German (see Table 4) with BAM outperforming standard CPT both in terms of German (0.7%) and English (1.0%) scores with minimal experience replay. Using our approximate experience replay and injecting German IFT data, further improves performance (3.5% in German and 4% in English), surpassing the base LLAMA-3-8B model now in both German and English benchmarks.

6.2 BAM for Instruction Fine-Tuning

We investigate the effectiveness of BAM for instruction finetuning, reporting results in Tables 3 and 4. We observe that BAM slightly improves learning of both German and Bulgarian, while significantly reducing forgetting. Considering a wider range of settings in Table 5, we observe that BAM with $N = K = 2$ and i.i.d. data slices not only strictly outperforms standard IFT on the combined data (IFT full) and an equal mix of Bulgarian and English data (IFT 50-50), but also IFT on just English data (IFT EN). Slicing the data by language (BAM BG | EN) results in even greater improvements and outperforms LLAMA-3-8B-Instruct (AI@Meta, 2024a) in both Bulgarian (10.8%) and English (1.3%). We hypothesize that merging the task vectors of IFT on multiple languages removes a lot of language-specific errors leaving a higher quality instruction following task vector.

6.3 Ablations

Below, we investigate various components and design decisions underlying BAM using the domain adaptation to Bulgarian.

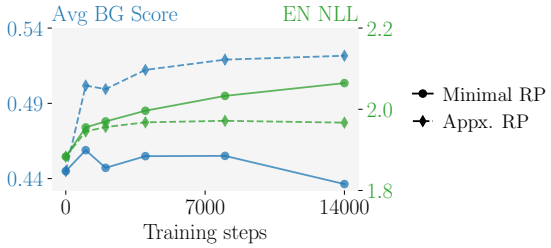

Figure 3: Comparing minimal and our approximate experience replay on MISTRAL with respect to average Bulgarian benchmark scores (↑) and Negative Log-Likelihood (NLL) on the English validation set (↓).

Approximate Experience Replay We compare our approximate experience replay, described in Section 4.1, to minimal experience replay, described in Section 4.2, for continued pretraining in Fig. 3. We observe that using minimal replay (solid lines in Fig. 3), target language performance (Avg BG Score – blue) first increases before dropping

Table 3: Effect of BAM with $N = 8$ and $K = 2$ for the language transfer to Bulgarian. We report normalized accuracies and their averages with full results on English benchmarks deferred to Table 8.

Model	CPT	IFT	Bulgarian Benchmarks												Avg BG	Avg EN	BG NLL	EN NLL
			WG	HS	ARC-c	ARC-e	MMLU	MathQA	GSM8K	TrQA	MON	Bele	XNLI	EXAMS				
L-8B	LLAMA-3-8B (Base)		61.48	52.19	34.89	53.32	50.81	35.37	36.99	27.19	40.91	45.77	45.46	45.75	44.18	63.85	1.695	**2.042**
	Standard	-	67.40	**66.48**	42.32	62.37	53.02	**38.65**	54.81	**38.69**	46.29	**62.11**	**48.75**	**56.43**	53.11	64.84	**1.018**	2.138
	Half LR	-	68.67	65.97	40.36	62.54	53.71	37.89	56.79	37.96	46.71	60.89	47.51	52.60	52.63	65.06	1.055	2.098
	BAM	-	**69.92**	66.14	**42.66**	**63.17**	**54.29**	37.48	**59.43**	38.53	**46.92**	59.66	48.03	52.60	**53.40**	**66.24**	1.061	2.097
	LLAMA-3-8B-Instruct		58.64	48.91	34.47	50.88	49.71	33.63	55.80	26.79	40.65	64.00	44.74	45.48	46.14	68.72	1.950	2.307
	BAM	Standard	68.67	66.75	47.95	**70.24**	52.54	38.73	63.84	31.70	48.60	**80.44**	50.92	51.51	55.99	67.69	1.208	2.290
	BAM	BAM	**68.98**	**68.01**	**49.57**	69.07	**54.04**	**38.56**	**65.05**	**36.17**	**49.94**	79.22	**51.45**	**53.42**	**56.96**	**69.97**	**1.148**	**2.193**
M-7B	MISTRAL-7B (Base)		61.48	53.63	37.54	55.93	49.37	31.36	29.04	29.32	42.15	39.67	42.77	44.93	43.10	59.81	1.525	**1.883**
	Standard	-	68.19	67.20	41.13	57.95	52.41	33.87	**65.73**	42.08	46.85	51.44	45.10	53.97	52.16	62.03	**1.408**	1.967
	BAM i.i.d.	-	69.77	**67.66**	41.04	60.01	**53.66**	34.61	58.23	41.78	45.60	52.67	47.11	53.15	52.11	**63.72**	1.411	1.951
	BAM	-	**70.24**	67.45	**43.26**	**61.62**	52.63	**35.58**	59.97	**42.24**	**46.98**	**52.78**	**48.23**	**54.79**	**52.98**	63.53	1.426	1.950

Table 4: Effect of BAM with $N = 4$ and $K = 2$ for the language transfer of LLAMA-3-8B to German. We report normalized accuracies and their averages with full results on English benchmarks deferred to Table 9.

CPT	IFT	German Benchmarks							Avg DE	Avg EN
		ARC-c	HellaSwag	MMLU	TruthfulQA	MGSM-DE	PAWS-DE	BeleBele		
LLAMA-3-8B (Base)	-	46.62	62.03	55.18	46.51	42.00	36.15	81.22	52.82	63.85
Standard min. replay	-	47.98	**66.49**	55.23	46.87	41.20	37.80	79.00	53.51	60.79
BAM min. replay	-	47.21	65.78	55.62	47.25	44.40	**39.80**	79.44	54.22	61.75
BAM appx. replay	-	**51.92**	65.97	**55.73**	**54.33**	**58.80**	35.35	**81.67**	**57.68**	**65.79**
BAM appx. replay	Standard	**53.12**	65.51	54.60	**55.20**	**66.00**	39.75	**86.44**	60.09	67.90
BAM appx. replay	BAM	52.95	**67.53**	**55.80**	54.28	65.60	**40.40**	85.89	**60.35**	**70.14**

Table 5: BAM for Bulgarian instruction tuning of our BAM trained LLAMA-3-8B.

Method	Avg BG	Avg EN
Base (BAM trained)	53.40	66.24
IFT full	55.99	67.69
IFT full - double BS	56.33	68.88
IFT full - half LR	56.37	68.97
IFT 50-50	55.01	67.55
IFT EN	54.72	67.76
IFT BG	54.16	66.96
BAM i.i.d.	56.45	68.65
BAM BG ∣ EN	**56.96**	**69.97**
LLAMA-3-Instruct	46.14	68.72

Table 6: Effect of approximate and minimal replay on source and target domain performance. BAM is with i.i.d. data slices.

Model	Language	Replay	CPT	Avg BG	Avg DE	Avg EN
LLAMA-3-8B	DE	min	Standard	-	53.51	60.80
			BAM	-	54.22	61.75
		appx	BAM	-	**57.68**	**65.79**
MISTRAL-7B	BG	min	Standard	43.71	-	51.44
			BAM	46.23	-	54.52
		appx	Standard	**52.16**	-	62.03
			BAM	52.11	-	**63.72**

off as capabilities of the base model are forgotten (increasing negative log-likelihood – green). In contrast, using our approximate experience replay (dashed line), we see a much stronger increase in target domain performance and reduced forgetting of the source domain. We confirm these findings in German (see Table 6) and thus use approximate experience replay for all other experiments.

BAM and Experience Replay We compare the effectiveness of BAM in the presence of minimal and approximate experience replay in Table 6 on Bulgarian, German, and English benchmarks. We observe that BAM is even more effective in the minimal replay setting, where the larger distribution shift induces more forgetting. There, BAM can, e.g., improve the performance in Bulgarian and English by 2.5% and 2.9%, respectively, compared to 0.0% and 1.7%, respectively, in the approximate replay setting.

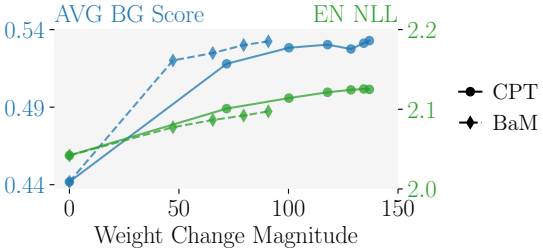

Figure 4: Average Bulgarian benchmark score (↑) and English NLL (↓) over L2 norm of weight change depending on training method for LLAMA-3

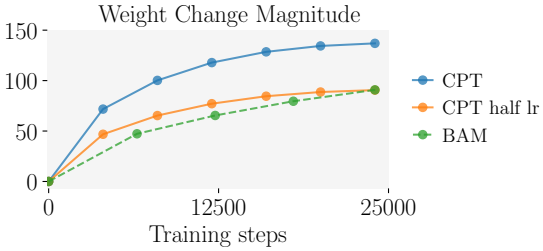

Figure 5: L2 norm of weight change depending on training method for LLAMA-3

Forgetting and Weight Change Magnitude We plot the average BG score as a measure of learning

and English NLL as a measure of forgetting over weight change magnitude in Fig. 4. We observe that both forgetting and learning strongly correlated with weight change magnitude and that BAM is more efficient, i.e., yields more learning and less forgetting at the same weight change, confirming our intuition discussed in Section 3.

Comparing BAM to standard CPT with a halved learning rate, we observe almost identical weight change magnitudes (see Fig. 5) corresponding to 66% of the standard CPT weight change. While the reduced learning rate CPT also reduces forgetting (although slightly less than BAM), it comes at the cost of severely reduced learning (see Table 3). We observe a similar but stronger effect for LORA which only shows minimal learning (see Table 12).

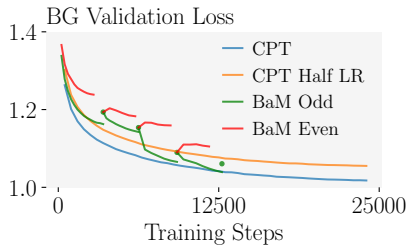

Figure 6: Bulgarian validation loss over training steps for LLAMA-3 depending on training method. BAM Odd (green) is trained on more Bulgarian and BAM Even (red) on more approximate experience replay. We show their merges as green dots.

Training Dynamics with BAM We compare the training dynamics of BAM and standard CPT at full and half learning rate in Fig. 6. We observe that training on data slices with larger portions of experience replay (even – red) cannot decrease Bulgarian validation loss further after a short period. However, after a merge, training on the Bulgarian-focused slices (odd – green) converges significantly faster than for CPT at a similar validation loss, highlighting the potential of merging to escape local minima or flatter portions of the loss landscape.

Table 7: Effect of of slice count N, parallelism factor K, and merging method on continued pertaining (CPT) of LLAMA-3 on a reduced Bulgarian dataset.

Merging Method	N	K	Avg BG	Avg EN	BG NLL	EN NLL
-	base		44.18	63.85	1.695	2.042
-	1	1	51.76	66.33	**1.136**	2.093
SLERP	2	2	**52.01**	**67.00**	1.194	2.077
	4	2	51.88	66.80	1.186	2.078
	4	4	51.25	66.76	1.233	**2.068**
LINEAR	4	2	51.98	66.65	1.186	2.078
MODEL STOCK	4	2	51.54	66.98	1.201	2.069

Effect of the Parallelism Factor We investigate the effect of the parallelism factor K on a dataset of 26 B tokens, obtained by combining the first two data slices $\mathcal{X}_1$ and $\mathcal{X}_2$, reporting results in Table 7 where all settings use the same data and compute. We observe that training on all data jointly ($N = 1, K = 1$) reduces Bulgarian NLL the most but at the cost of increased forgetting (highest English NLL) leading to worse benchmark performance than BAM. Comparing BAM hyperparameters, we observe that increasing K reduces both learning and forgetting as we reduce weight change magnitude and improve task vector quality ($N = 4, K = 4$). The best trade-off leading to the highest English and Bulgarian scores is attained with a parallelism factor of $K = 2$ and data slices of roughly 10B tokens ($N = 2, K = 2$). We thus choose these settings for all other experiments leading to $N \in \{4, 8\}$ for the full data.

Effect of the Merging Method We compare SLERP, LINEAR, and MODEL STOCK merging in Table 7 and observe that SLERP and LINEAR merging achieve almost identical results, with MODEL STOCK reducing forgetting at the cost of reduced learning. As SLERP merging achieves slightly better scores, we use it for all other experiments.

Sensitivity to Hyperparameters Varying batch size, learning rate, and weight decay in the IFT stage, we observe that the performance of standard IFT is highly sensitive to these hyperparameter choices. In contrast, BAM IFT is robust across the same parameter ranges. In Table 5, we find that doubling the batch size or halving the learning rate for standard IFT may help reduce catastrophic forgetting and even improve downstream performance similar to BAM i.i.d. However, BAM does not require an extensive and costly search for optimal hyperparameters and still outperforms all other approaches when splitting the data by language.

7 Related work

Catastrophic Forgetting Neural networks trained on a specific task are known to catastrophically forget the previous task when adapted to a new one (French, 1999; Goodfellow et al., 2014; Kemker et al., 2018). While this becomes less pronounced as model and pertaining data size grow (Ramasesh et al., 2022), it remains a severe issue even for modern LLMs (Zhai et al., 2023; Shi et al., 2024; Li and Lee, 2024; Gogoulou et al., 2023).

Mitigating Catastrophic Forgetting As LLMs are frequently finetuned or continually pretrained on new tasks, mitigating catastrophic forgetting has become essential and a wide range of methods has been proposed. Lee et al. (2020) suggest to randomly reset weights to their pretrained state. Biderman et al. (2024) show that LoRA reduces forgetting at the cost of reduced learning. Huang et al. (2024) suggest experience replay with synthetic and Ibrahim et al. (2024) with original source domain samples. Winata et al. (2023) propose to exponentially reduce the learning rate when learning new tasks. Similar to us, Lin et al. (2024) suggest to linearly merge the adapted with the original model using block-wise parameters, focusing on alignment tuning instead of language transfer.

Model Merging Model merging was originally proposed in federated learning (McMahan et al., 2017) to lower communication costs and successfully deployed (Stoica et al., 2023; Matena and Raffel, 2022). As a way to combine multiple models without training, it has recently gained popularity in the LLM community (Goddard et al., 2024). Popular methods include Linear or Task Arithmetic (Ilharco et al., 2023), which perform linear interpolation of task vectors, its extension Model Breadcrumbs (Davari and Belilovsky, 2023), which discards large weight changes, Ties (Yadav et al., 2023), which uses heuristics favoring large weight changes, Dares (Yu et al., 2023), which randomly drops weight changes before merging, Model Stock (Jang et al., 2024), which merge weights layer-wise, to in expectation, minimize distance to the center of the task vector distribution, and Slerp (Shoemake, 1985), which averages weights in polar coordinates.

Multiple works have shown that merging during pretraining or finetuning, especially on non-IID data, can match or improve the performance of compound training. Wortsman et al. (2023) average models finetuned without any communication. Li et al. (2022) propose to iteratively branch and merge models during training, however, they assume the full data distribution is available for pertaining and focus on building ensembles rather than a single model. Cold Fusion (Don-Yehiya et al., 2023) is most similar to our work but focuses on training a base model that can then be easily adapted to a new task, rather than this adaptation itself. This objective is shared by Choshen et al. (2022) which only merge once.

8 Conclusion

We proposed Branch-and-Merge (BAM) training to mitigate forgetting while boosting learning in language transfer by generating lower magnitude but higher quality weight changes. We showed that combining BAM with an effective approximate experience replay data mix significantly reduces forgetting. Finally, we demonstrated that our approach can benefit both continuous pertaining and instruction tuning in both alphabet-sharing (German) and non-sharing (Bulgarian) languages. For instance, we outperform Llama-3-8B-Instruct with the same base model in both source (English, 1.3%) and target (Bulgarian, 10.8%) languages.

9 Limitations

Our study focuses on language transfer to two languages with different characteristics and considers two models of up to 8 billion parameters. However, to establish the general applicability of our approach, potentially even to general domain adaptation, further experiments across a broader set of languages and tasks as well as model architectures will be necessary.

We considered specific data mixes for the continued pretraining in both considered languages which we observe to yield good performance — it is possible that the success of Branch-and-Merge depends on the composition of these datasets. While infeasible when adapting state-of-the-art pretrained models with unknown training set distribution, an evaluation of our method with exact experience replay would provide further understanding of its performance relative to the state-of-the-art in continuous learning, including joint training on all data.

While we consider a broad range of up to 12 benchmarks per language, they are still limited in their domain coverage. As BAM does not outperform standard training across all benchmarks, this benchmark composition can affect the resulting conclusions.

While we originally optimized hyperparameters for standard training and carried them over to BAM, it is possible, although unlikely, that a more extensive hyperparameter search would benefit standard training more than BAM. Empirically, we find for IFT that BAM is more robust to hyperparameter changes than standard IFT.

10 Ethical Considerations

We believe our work empowers practitioners to more efficiently adapt strong pretrained models to other potentially low-resource languages, thus contributing to the democratization of large language models. However, such models can of course also be abused and in particular if our approach generalizes beyond language to general domain adaptation, malicious practitioners could more efficiently adapt the models for nefarious tasks.

Acknowledgments

This research was partially funded by the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure).

This work has been done as part of the EU grant ELSA (European Lighthouse on Secure and Safe AI, grant agreement no. 101070617). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

The work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

We would like to thank Dr. Maurice Weber for helping adapt the RPv2 pipeline to Bulgarian and create the Bulgarian dataset. We would also like to thank Hristo Venev for his help with system related issues.

References

AI@Meta. 2024a. Llama 3 instruct details.

AI@Meta. 2024b. Llama 3 model card.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Vít Baisa, Jan Michelfeit, Marek Medveď, and Miloš Jakubíček. 2016. European Union language resources in Sketch Engine. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2799–2803, Portorož, Slovenia. European Language Resources Association (ELRA).

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *Preprint*, arXiv:2308.16884.

Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. 2024. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*.

S. Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation.

Zhihong Chen, Shuo Yan, Juhao Liang, Feng Jiang, Xiangbo Wu, Fei Yu, Guiming Hardy Chen, Junying Chen, Hongbo Zhang, Li Jianquan, Wan Xiang, and Benyou Wang. 2023. MultilingualSIFT: Multilingual Supervised Instruction Fine-tuning.

Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. 2022. Fusing finetuned models for better pretraining. *Preprint*, arXiv:2204.03044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

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*, abs/2110.14168.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Yiming Cui and Xin Yao. 2024. Rethinking LLM language adaptation: A case study on chinese mixtral. *CoRR*, abs/2403.01851.

Luigi Daniele and Suphavadeeprasit. 2023. Amplify-instruct: Synthetically generated diverse multi-turn conversations for efficient llm training. *arXiv preprint arXiv:(coming soon)*.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*.

MohammadReza Davari and Eugene Belilovsky. 2023. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *CoRR*, abs/2312.06795.

Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, and Leshem Choshen. 2023. ColD fusion: Collaborative descent for distributed multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 788–806, Toronto, Canada. Association for Computational Linguistics.

Tomaž Erjavec, Maciej Ogrodniczuk, Petya Osenova, et al. 2023. The parlamint corpora of parliamentary proceedings. *Language Resources and Evaluation*, 57(2):415–448.

Wikimedia Foundation. Wikimedia downloads.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Hao Fu, Yao; Peng and Tushar Khot. 2022. How does gpt obtain its ability? tracing emergent abilities of language models to their sources. *Yao Fu's Notion*.

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, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Leonidas Gee, Andrea Zugarini, Leonardo Rigutini, and Paolo Torroni. 2022. Fast vocabulary transfer for language model compression. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 409–416, Abu Dhabi, UAE. Association for Computational Linguistics.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's mergekit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*.

Evangelia Gogoulou, Timothée Lesort, Magnus Boman, and Joakim Nivre. 2023. A study of continual learning under language shift. *CoRR*, abs/2311.01200.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. 2014. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Olmo: Accelerating the science of language models. *CoRR*, abs/2402.00838.

Momchil Hardalov, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nakov. 2020. EXAMS: A multi-subject high school examinations dataset for cross-lingual and multilingual question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5427–5444, Online. Association for Computational Linguistics.

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

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*.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. *CoRR*, abs/2403.01244.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *Submitted to Transactions on Machine Learning Research*. Under review.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Neel Jain, Ping yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. NEFTune: Noisy embeddings improve instruction finetuning. In *The Twelfth International Conference on Learning Representations*.

Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. 2024. Model stock: All we need is just a few fine-tuned models. *CoRR*, abs/2403.19522.

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. *CoRR*, abs/2310.06825.

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Victoria Lin, Wen-tau Yih, and Srinivasan Iyer. 2024. Instruction-tuned language models are better knowledge learners. *CoRR*, abs/2402.12847.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler L. Hayes, and Christopher Kanan. 2018. Measuring catastrophic forgetting in neural networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3390–3398. AAAI Press.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020. Mixout: Effective regularization to finetune large-scale pretrained language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chen-An Li and Hung-Yi Lee. 2024. Examining forgetting in continual pre-training of aligned large language models. *CoRR*, abs/2401.03129.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022. Branch-train-merge: Embarrassingly parallel training of expert language models. In *First Workshop on Interpolation Regularizers and Beyond at NeurIPS 2022*.

Wing Lian, Guan Wang, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yüksekgönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models. *CoRR*, abs/2211.09110.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the alignment tax of rlhf. *Preprint*, arXiv:2309.06256.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu.

Yingwei Ma, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. 2023. At which training stage does code data help llms reasoning? *CoRR*, abs/2309.16298.

Michael Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

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.

Vladislav Mosin, Igor Samenko, Borislav Kozlovskii, Alexey Tikhonov, and Ivan P. Yamshchikov. 2023. Fine-tuning transformers: Vocabulary transfer. *Artif. Intell.*, 317(C).

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

Galileo Mark Namata, Ben London, Lise Getoor, and Bert Huang. 2012. Query-driven active surveying for collective classification. In *International Workshop on Mining and Learning with Graphs*, Edinburgh, Scotland.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2024. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4226–4237, Torino, Italia. ELRA and ICCL.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z.

Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon LLM: outperforming curated corpora with web data only. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Björn Plüster. 2023a. German Benchmark Datasets.

Björn Plüster. 2023b. Leolm/openschnabeltier dataset.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020a. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020b. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2022. Effect of scale on catastrophic forgetting in neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 348–358.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 6107–6122. Association for Computational Linguistics.

Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Zhengzhong Liu, Hongyi Wang, Bowen Tan, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric P. Xing. 2023. Slimpajama-dc: Understanding data combinations for LLM training. *CoRR*, abs/2309.10818.

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*.

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. *CoRR*, abs/2404.16789.

Ken Shoemake. 1985. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1985, San Francisco, California, USA, July 22-26, 1985*, pages 245–254. ACM.

George Stoica, Daniel Bolya, Jakob Bjorner, Taylor Hearn, and Judy Hoffman. 2023. Zipit! merging models from different tasks without training. *CoRR*, abs/2305.03053.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari Morcos. 2023. D4: improving LLM pretraining via document de-duplication and diversification. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Together.ai. 2023. Redpajama: an open dataset for training large language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Tamás Váradi, Bence Nyéki, Svetla Koeva, Marko Tadić, Vanja Štefanec, Maciej Ogrodniczuk, Bartłomiej Nitoń, Piotr Pęzik, Verginica Barbu Mititelu, Elena Irimia, Maria Mitrofan, Dan Tufiș, Radovan Garabík, Simon Krek, and Andraž Repar. 2022. Introducing the CURLICAT corpora: Seven-language domain specific annotated corpora from curated sources. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 100–108, Marseille, France. European Language Resources Association.

Genta Indra Winata, Lingjue Xie, Karthik Radhakrishnan, Shijie Wu, Xisen Jin, Pengxiang Cheng, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2023. Overcoming catastrophic forgetting in massively multilingual continual learning. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 768–777. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Mitchell Wortsman, Suchin Gururangan, Shen Li, Ali Farhadi, Ludwig Schmidt, Michael Rabbat, and Ari S. Morcos. 2023. lo-fi: distributed fine-tuning without communication. *Transactions on Machine Learning Research*.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak 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 *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. 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 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2023. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *CoRR*, abs/2311.03099.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

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*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. *CoRR*, abs/2309.10313.

Zihan Zhang, Meng Fang, Ling Chen, and Mohammad-Reza Namazi-Rad. 2023. CITB: A benchmark for continual instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9443–9455. Association for Computational Linguistics.

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer. *CoRR*, abs/2401.01055.

A Extended Evaluation

Data Slice Order We evaluate the effect of data slice order on LLAMA-3 in Table 10. We observe that reversing the order of data slices for BAM training has only a minimal effect on both forgetting and domain adaptation. Interestingly, merging the two models obtained via these different orders is strictly better than either, although at twice the computational cost. This highlights again the effectiveness of BAM at finding optimal task vectors by merging out the error component.

Tokenizer Extension A common challenge with LLM domain adaptation is that the LLM's tokenizer may not be well suited for the target domain, expressed in a higher fertility. This entails longer training, slower inference, shorter effective context length as well as potential performance degradation.

In our language transfer setting from English to Bulgarian, we also make use of tokenizer/vocabulary expansion for some of our experiments to reduce the computational cost. In the case of MISTRAL-7B, we find that Bulgarian tokenization is subpar. To this end, we train a SentencePieceBPE (Kudo and Richardson, 2018) tokenizer with a vocabulary of 8k tokens on high-quality Bulgarian text. We find that a mix of 75% RPv2 BG and 25% Wikipedia, where the whole Bulgarian Wikipedia comprises these 25% gave the lowest fertility on a sample from mC4 (Xue et al., 2021). After removing all tokens that do not include at least one Cyrillic character or are already in the original tokenizer, we are left with exactly 6000 new tokens, which are then appended to the original Mistral tokenizer with their respective SentencePiece scores. This whole procedure ensures that the English tokenization remains practically unchanged, which is important to reduce Catastrophic Forgetting. We initialize the new input and output embeddings with their mean tokenization using the original tokenizer and add them to the model's vocabulary in the style of VIPI (Mosin et al., 2023) and FVT (Gee et al., 2022). We report results for MISTRAL-7B in Table 11 and use an 8k (effectively 6k) tokenizer extension for all further MISTRAL-7B experiments due to the greatly increased training and inference efficiency at very similar performance and retain the original LLAMA-3-8B tokenizer due to its already huge vocabulary and lower fertility. Note: Reducing or increasing the amount of Web data in that tokenizer training mix resulted in higher fertility on the mC4 sample. The reason for this is not fully clear and we intend on investigating this in future work.

Low-Rank Adaptation LoRA has become widely popular as a method for cheaper finetuning of LLMs (Hu et al., 2022). Taking into consideration the contribution of (Biderman et al., 2024), which puts LoRA in the context of learning less but also forgetting less we also show how LoRA fairs in our Language Transfer setting. Due to limited compute resources, we do not perform an extensive hyperparameter sweep and instead copy what we can from the Code CPT experiment in Biderman et al. (2024). As far as we know the batch sizes are not mentioned there and we decide to stick to 512, while deducting that the original may have used 128. We also proportionately increase the learning rate and find that $4e-5$ converges the fastest. The comparison in Table 12 is in the reduced, 20B-token setting, same as in Table 7. We indeed observe a better preservation of the English Negative Log Likelihood but also a significant reduction in learned Bulgarian capabilities. It may be the case that the Language Transfer adaptation is not as low-rank as it is for Code and the referred LoRA rank parameter should be set higher than 256.

B Benchmark Details

B.1 Benchmark Descriptions

Below we provide short descriptions of all datasets and note the license they are published under. German language benchmarks is run in a 5-shot setting. For the other evaluations, we specify the number of shots below, or use 0-shots when not specified.

HellaSwag (MIT License) (Zellers et al., 2019) is a common sense reasoning benchmark asking an LLM to select a logical continuation of a sentence. Evaluated on the 10000 sample validation set.

Winogrande (Appache 2.0 License)(Sakaguchi et al., 2021) is a common sense reasoning benchmark asking an LLM to fill in a blank from a choice of two entities to logically complete a sentence. Evaluated on the 1767 sample validation set of `winogrande_xl`.

ARC-Easy and -Challenge (CC BY-SA License) (Clark et al., 2018) is a dataset of science exam questions. Evaluated on the 2590 hard

Table 8: English Benchmark performance of LLAMA-3-8B continuously pretrained on Bulgarian

Training	WG	HS	ARC-c	ARC-e	MMLU	Bele	MathQA	GSM8K	TrQA	AVG
Base	73.00	79.12	**53.32**	**77.65**	**65.16**	66.77	40.00	47.99	**71.62**	63.85
CPT	74.43	80.00	50.34	71.59	61.84	**71.33**	38.29	**71.95**	63.79	64.84
BAM	**74.58**	**80.12**	52.81	75.54	63.25	71.11	**41.07**	69.97	67.65	**66.23**

Table 9: English Benchmark performance of LLAMA-3-8B continuously pretrained on German

Model	WG	HS	ARC-c	ARC-e	MMLU	Bele	MathQA	GSM8K	TrQA	AVG
Base	**73.00**	**79.12**	**53.32**	**77.65**	**65.16**	**66.77**	**40.00**	**47.99**	**71.62**	**63.85**
CPT	72.45	78.80	51.19	77.60	62.90	55.88	39.27	41.16	67.78	60.79
BAM	72.84	78.65	50.85	77.02	63.87	58.66	39.83	44.73	69.23	61.74

Table 10: Effect of data slice order on BAM.

Data Order	Avg EN	Avg BG	BG NLL	EN NLL
Base Model	63.85	44.18	1.695	2.042
Standard	66.23	53.06	**1.061**	2.097
Reversed	65.64	52.70	1.167	2.071
Merged	**66.26**	**53.34**	1.076	**2.069**

Table 11: Effect of tokenizer extension on performance before and after continuous pertaining (CPT) of MISTRAL-7B.

Training	Extension	Fertility	Avg BG	Avg EN
Base	None	2.37	44.50	63.50
	8k	**1.71**	29.28	62.57
CPT	None	2.37	**51.47**	**61.48**
	8k	**1.71**	50.93	60.96

Table 12: Effect of LoRA regularization compared to BAM on LLAMA-3.

Training	Avg BG	Avg EN	BG NLL	EN NLL
Base	44.18	63.85	1.695	2.042
CPT	51.76	66.33	**1.136**	2.093
BAM	**52.01**	**67.00**	1.194	**2.077**
LoRA	45.33	64.71	1.515	2.059

sample (ARC-Challenge) and 5197 easy samples (ARC-Easy).

MMLU (MIT License) (Hendrycks et al., 2021) is a multitask language understanding benchmark covering a wide range of 57 different tasks. Evaluated on 14079 test set samples. We evaluate MMLU using 5-shots.

GSM8K (MIT License) (Cobbe et al., 2021) is a mathematical reasoning benchmark consisting of grade-school math questions for which free text answers must be provided. Evaluated on 1.3k test set samples. We run GSM8k with 8-shot chain-of-thought generation.

MathQA (Apache 2.0 License) (Amini et al., 2019) is a multiple choice mathematical reasoning benchmark. Evaluated on 4475 validation set samples.

Belebele (CC-BY-NC 4.0 License) (Bandarkar et al., 2023) is a multiple choice reading comprehension dataset. Evaluated on 900 samples per language.

TriviaQA (Apache 2.0 License) (Joshi et al., 2017)) is trivia question dataset. Evaluated on 17.9k validation set samples. We use 5-shot evaluation.

XNLI (CC BY-NC 4.0 License) (Conneau et al., 2018) is a language understanding dataset where the task is to decide whether two statements contradict one-another, are neutral, or one entails the other. Evaluated on 2.5k validation samples.

EXAMS (CC BY-SA 4.0 License) (Hardalov et al., 2020) is a high school exam question dataset covering a range of subjects. Evaluated 1472 test set samples in Bulgarian. We use 5-shot evaluation.

PAWS (Special License permitting "free use for any purpose") (Yang et al., 2019) is a reading comprehension dataset where the task is to decide whether two benchmarks are paraphrases. Evaluated on 1967 test set samples.

MGSM (CC BY 4.0 License) (Shi et al., 2023) is mathematical reasoning benchmark manually translated from GSM8k. Evaluated on 250 test set samples.

B.2 Bulgarian Benchmarks

Translation We use Google Translate to machine translate the text of the benchmark problems and answers. Additional, we identified a set of heuristics for cases where the machine translation is of low quality, such as inconsistent translations of the same word and not following exact format in both source and target sentences. In all such cases, we gave the tasks to human translators with additional instructions on possible problems we identified in each benchmark. Overall human translators manually translated 2143 test set samples.

A notable example of a benchmark with significant problems that we expect to repeat in many other languages is Winogrande challenge (Sakaguchi et al., 2021). In this case, one of two words have to be chosen based on world knowledge and reasoning. However, with machine translation or naïve human translation to non-English, the actual answer can be revealed in a much easier way by the means of having only one answer that is in gender agreement with other words in the sentence. We performed manual translations that used synonyms that do not exhibit such behavior and as a result, the translated benchmark is not easier than the original. The translated versions of the benchmarks with these fixes are made publicly available.

MON The MON dataset is obtained as private data from the Bulgarian Ministry of Education. This contains 10088 exam questions with 4 possible choices, only one of which is correct, spanning topics from 4th to 12th grade tests previously given for external tests to schools in Bulgaria. The questions span all subjects tested by the official Bulgarian curriculum but exclude problems such as geometry tasks that include images in their problem definition or answers. The dataset is not publicly available and as a result, we expect it to be less likely to be in any of the training data in any form.

C Dataset Details

C.1 IFT Set Composition

We make note of the good performance and instruction following capabilities of the Intel Neural-Chat models and decide to include SlimOrca (Lian et al., 2023; Mukherjee et al., 2023; Longpre et al., 2023) and MetaMathQA (Yu et al., 2024) in our English IFT data mix. To fill in the gap of multi-turn conversation data we additionally include the Capybara dataset (Daniele and Suphavadeeprasit, 2023),

Table 13: Composition of the Bulgarian IFT dataset.

Dataset	Domain	#Examples	Repetitions	Prob [%]
OpenHermes-2.5-BG	Mixed Conversations	50,000	1	64.10
Capybara-BG	Mixed Conversations	16,000	1	20.51
MetaMath-BG	Math	10,000	1	12.82
CodeAlpaca-BG	Code	2,000	1	2.56

Table 14: Composition of the English IFT dataset.

Dataset	Domain	#Examples	Repetitions	Prob [%]
SlimOrca	Mixed Conversations	517,982	1	55.76
MetaMathQA	Math	395,000	1	42.52
Capybara	Mixed Conversations	16,000	1	1.72

which we have observed from our experience boost the models' "chattiness" and overall response quality.

The fact that there are no publicly available general Bulgarian IFT datasets, lead us to the translation of already existing ones. We use machine translation to produce 50K Bulgarian translated samples from the OpenHermes-2.5 (Teknium, 2023) dataset, 10K samples from MetaMathQA (Yu et al., 2024) and 2K samples of code with Bulgarian instructions from CodeAlpaca (Chaudhary, 2023). We take special care in the translation of the Capybara (Daniele and Suphavadeeprasit, 2023) and OpenHermes datasets. Through a combination of classification and manual inspection, we identify examples, where the machine translation is not good enough to make a sensible training example, e.g. instructions that require rhyming, as the words that rhyme in English will most likely not rhyme in Bulgarian. The identified 5% of the Capybara dataset is then manually translated/adjusted to fit the Bulgarian language. See Table 16 for full details and licenses.

C.2 Validation Set Composition

Constructing validation datasets for language model training, especially when such are trained on web-crawl data, is a challenging task with respect to avoiding data contamination. Our Bulgarian validation set consists of a total of 40K examples, 30K of which are a held-out set of news articles from a specific media outlet and the other 10K is a mix of dialogs, questions and answers, literary works and legal documents. The English validation dataset is comprised of 25K random samples from the FineWeb-Edu dataset (Lozhkov et al., 2024) 7K samples from arXiv scientific papers, 3K from the PubMed dataset (Namata et al., 2012) and 5K

Table 15: Composition of the German IFT dataset.

Dataset	Domain	#Examples	Repetitions	Prob [%]
evol-instruct-deutsch	Mixed Conversations	59,022	1	45.15
alpaca-gpt4-deutsch	Mixed Conversations	50,000	1	38.23
OpenSchnabeltier	Mixed Single-turn	21,749	1	16.62

Table 16: Sources and licenses of used datasets

Dataset	Source	License
RPv2 pipeline	Together.ai (2023)	Apache 2.0
OpenWebText	Gokaslan et al. (2019)	CC0-1.0
CulturaX	Nguyen et al. (2024)	CC0-1.0
FineWeb-Edu	Lozhkov et al. (2024)	ODC-BY
PubMed	Namata et al. (2012)	Unknown
Eur-Lex	Baisa et al. (2016)	CC-BY-NC-SA
Wikipedia	Foundation	CC-BY-SA-3.0
OrcaMath	Mitra et al. (2024)	MIT
Parlamint	Erjavec et al. (2023)	CC-BY
OpenHermes-2.5	Teknium (2023)	Unknown
Capybara	Daniele and Suphavadeeprasit (2023)	Apache 2.0
Curlicat	Váradi et al. (2022)	CC-BY-SA-4.0
SlimOrca	Lian et al. (2023); Mukherjee et al. (2023)	MIT
CodeAlpaca	Chaudhary (2023)	CC-BY-4.0
Europarl	Koehn (2005)	Unknown
MetaMath	Yu et al. (2024)	MIT
Open-Platypus	Lee et al. (2023)	Apache 2.0
alpaca-gpt4-deutsch	Chen et al. (2023)	Apache 2.0
OpenSchnabeltier	Plüster (2023b)	Apache 2.0
evol-instruct-deutsch	Chen et al. (2023)	Apache 2.0

books from the Project Gutenberg[†].

D Experimental Setup and Evaliation Details

D.1 Training parameters

We use the same exact training hyperparameters for both MISTRAL-7B and LLAMA-3-8B based models. We stick to the 8192 size context lengths and train with sequence packing, without truncation. Based on prior work and initial experiments, we find that $1e-5$ is the best maximum learning rate for continued pre-training in our settings with a batch size of 512 for continued pre-training and 256 for supervised fine-tuning, effectively training for 4M and 2M tokens respectively. The optimizer in use is AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.95$ and 0.05 weight decay rate. We use a cosine decay learning rate scheduler, that decays the LR to $0.1 \cdot$ max_lr with $\max(100, 0.01 \cdot$ total_steps$)$ of linear warmup.

For fine-tuning, we have found that training for more than 2 epochs on a given IFT dataset with the aforementioned hyperparameters is not beneficial and exaggerates catastrophic forgetting. Additionally, we add embedding vector noise during training through NEFTune (Jain et al., 2024) with a noise-$\alpha = 5$. In this stage, we train only on the IFT completions and not on the prompts. This is important to prevent unwanted self-talking behavior in live usage.

[†] https://www.gutenberg.org/

Since we train on 64 GPUs at once, we exploit DeepSpeed ZeRO (Rasley et al., 2020; Rajbhandari et al., 2020b) stage 1 with mixed precision training in bf16. Combining this with activation checkpointing and FlashAttention-2 (Dao, 2024) allows us to use a batch size of 2 during training and evaluation. For reference, our setup allows the models to train with up to 7000 tokens per second per GPU.

D.2 Computational Budget

All model training and evaluations were conducted on a cluster of 64 NVIDIA H100 GPUS (8 nodes x 8 GPUs) with InfiniBand and 224 available CPU cores per node. The total computational cost of the experiments included in this paper, including exploratory ones not mentioned here, is around $80,000$ NVIDIA H100 GPU hours. The tokenizer extension we perform on MISTRAL-7B (Base) helps reduce the training and inference cost of our Mistral-based models by roughly 30%.

Table 17: Composition of the Bulgarian curriculum splits.

Split	# Total BG	# Total Replay	Dataset	Repetitions	Replay
$\mathcal{X}_1$	14.7B	850M	Wikipedia-BG	1	✗
			OpenWebText	0.1	✓
			Bulgarian Law	1	✗
			Eur-Lex-BG	1	✗
			IFT-BG	1	✗
			RPv2-BG	0.2	✗
$\mathcal{X}_2$	8.3B	3.3B	Wikipedia-EN	0.25	✓
			OpenWebText	0.15	✓
			GitHub repos	0.2	✓
			IFT-EN	1	✓
			RPv2-BG	0.12	✗
$\mathcal{X}_3$	11.4B	850M	Wikipedia-BG	1	✗
			OpenWebText	0.1	✓
			Bulgarian Law	1	✗
			Eur-Lex-BG	1	✗
			IFT-BG	1	✗
			RPv2-BG	0.12	✗
			Parlamint-BG	1	✗
			Europarl-BG	1	✗
			Legal docs	0.4	✗
$\mathcal{X}_4$	8.3B	3.3B	Wikipedia-EN	0.25	✓
			OpenWebText	0.15	✓
			GitHub repos	0.2	✓
			IFT-EN	1	✓
			RPv2-BG	0.12	✗
$\mathcal{X}_5$	12.4B	850M	Wikipedia-BG	1	✗
			OpenWebText	0.1	✓
			Bulgarian Law	1	✗
			Books	1	✗
			IFT-BG	1	✗
			RPv2-BG	0.1	✗
			Parlamint-BG	1	✗
			Europarl-BG	1	✗
			Legal docs	0.4	✗
$\mathcal{X}_6$	8.3B	3.3B	Wikipedia-EN	0.25	✓
			OpenWebText	0.15	✓
			GitHub repos	0.2	✓
			IFT-EN	1	✓
			RPv2-BG	0.12	✗
$\mathcal{X}_7$	10.3B	850M	Wikipedia-BG	1	✗
			OpenWebText	0.1	✓
			Bulgarian Law	1	✗
			Books	1	✗
			IFT-BG	1	✗
			RPv2-BG	0.1	✗
			Parlamint-BG	1	✗
			Europarl-BG	1	✗
			Legal docs	0.2	✗
$\mathcal{X}_8$	8.3B	3.7B	Wikipedia-EN	0.25	✓
			OpenWebText	0.15	✓
			GitHub repos	0.4	✓
			IFT-EN	1	✓
			RPv2-BG	0.12	✗