
Proceedings Paper:
Yamaguchi, A., Morishita, T., Villavicencio, A. et al. (2026) Mitigating catastrophic forgetting in target language adaptation of LLMs via Source-Shielded Updates. In: Liakata, M., Moreira, V.P., Zhang, J. and Jurgens, D., (eds.) Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026). 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026), 02-07 Jul 2026, San Diego, California. Association for Computational Linguistics (ACL), pp. 18949-18975. ISBN: 9798891763906.

Mitigating Catastrophic Forgetting in Target Language Adaptation of LLMs via Source-Shielded Updates

Atsuki Yamaguchi[1] **Terufumi Morishita**[2] **Aline Villavicencio**[1,3,4] **Nikolaos Aletras**[1]
[1]University of Sheffield, United Kingdom [2]Hitachi, Ltd., Japan
[3]University of Exeter, United Kingdom [4]Federal University of Rio Grande do Norte, Brazil
{ayamaguchi1,a.villavicencio,n.aletras}@sheffield.ac.uk

Abstract

Expanding the linguistic diversity of instruct large language models (LLMs) is crucial for global accessibility but is often hindered by the reliance on costly specialized target language labeled data and catastrophic forgetting during adaptation. We tackle this challenge under a realistic, low-resource constraint: adapting instruct LLMs using only unlabeled target language data. We introduce **S**ource-**S**hielded **U**pdates (**SSU**), a selective parameter update strategy that proactively preserves source knowledge. Using a small set of source data and a parameter importance scoring method, SSU identifies parameters critical to maintaining source abilities. It then applies a column-wise freezing strategy to protect these parameters before adaptation. Experiments across five typologically diverse languages and 7B and 13B models demonstrate that SSU successfully mitigates catastrophic forgetting. It reduces performance degradation on monolingual source tasks to just 3.4% (7B) and 2.8% (13B) on average, a stark contrast to the 20.3% and 22.3% from full fine-tuning. SSU also achieves target-language performance highly competitive with full fine-tuning, outperforming it on all benchmarks for 7B models and the majority for 13B models.[1]

1 Introduction

Large language models (LLMs) demonstrate remarkable generalization across numerous applications (OpenAI, 2025; Guo et al., 2025; Yang et al., 2025; Gemma Team et al., 2025). However, they notoriously underperform in languages absent or underrepresented in their training data, creating a barrier to equitable access for speakers worldwide (Huang et al., 2023). The standard approach to resolve this issue is continual pre-training (CPT) or fine-tuning on target language data (Cui et al., 2024; Ji et al., 2025).

[1]Our code and models are available via https://github.com/gucci-j/ssu.

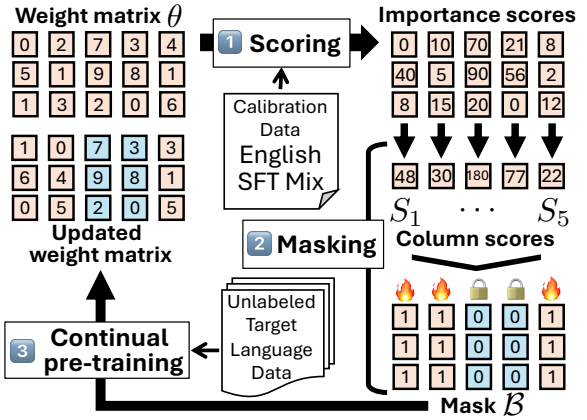

Figure 1: Overview of **S**ource-**S**hielded **U**pdate (**SSU**). The method comprises three stages: importance scoring, column-wise mask generation, and continual pre-training on unlabeled target data with the masks.

Yet, adapting instruct models to these languages is uniquely challenging. Such models require specialized instruction-tuning data (Wei et al., 2022; Rafailov et al., 2023), which is often unavailable or prohibitively costly to create for underrepresented languages (Huang et al., 2024c). Furthermore, machine-translated data as a low-cost alternative is not consistently effective (Tao et al., 2024).

Consequently, using unlabeled target language text is often the only viable option for adaptation. While this approach can improve target language proficiency, it often triggers catastrophic forgetting (Kirkpatrick et al., 2017; Tejaswi et al., 2024; Mundra et al., 2024; Yamaguchi et al., 2025), where new training erases prior knowledge. This issue is acute for instruct models, as it cripples the general-purpose functionality of the model, which is primarily derived from core abilities like chat and instruction-following. In response, previous work has attempted **post-hoc** mitigation. For example, Yamaguchi et al. (2025) merge weights of the original and adapted models, while Huang et al. (2024c) use a task vector and apply param-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 18949–18975

eter changes from CPT on the base model to the instruct model. Nonetheless, these methods largely fail to mitigate catastrophic forgetting, substantially degrading these core functionalities.

The shortcomings of post-hoc methods suggest that *mitigation should occur during adaptation.* We therefore focus on **the CPT stage**. Specifically, we leverage selective parameter updates, a method of restricting which weights are modified during training. This approach is proven more effective at mitigating catastrophic forgetting than alternatives like parameter-efficient fine-tuning, regularization, or model merging (Zhang et al., 2024a; Hui et al., 2025). However, existing selective parameter tuning paradigms for adapting LLMs are ill-suited for adapting instruct models with unlabeled target language text. They rely either on **random selection**, offering no principled way to preserve knowledge, or on signals from the new data to guide updates (**target-focused**) (§2). Target-focused signals are particularly vulnerable because raw text lacks chat templates required to elicit instruction-following behavior. Optimizing for this incompatible format risks corrupting the very foundational capabilities we aim to preserve due to the structural differences between raw text and chat templates.

We therefore introduce **S**ource-**S**hielded **U**pdates (**SSU**), a novel **source-focused** approach that *proactively shields source knowledge before adaptation begins* (Figure 1). First, SSU identifies parameters critical to source abilities using a small set of source data and a parameter importance scoring method, such as those used in model pruning, e.g., Wanda (Sun et al., 2024). Second, it uses these element-wise scores to construct a column-wise freezing mask. This structural design is crucial. Unlike naive element-wise freezing that corrupts feature transformations, our column-wise approach preserves them entirely. Finally, this mask is applied during CPT on unlabeled target language data, keeping the shielded structural units frozen. This process allows SSU to effectively preserve the general-purpose ability of the model while improving target language performance.

We verify our approach through extensive experiments with five typologically diverse languages and two different model scales (7B and 13B). We evaluate source language (English) performance across dimensions including chat, instruction-following, safety, and general generation and classification, alongside target language performance. We summarize our contributions as follows:

- A novel method for adapting instruct models to a target language without specialized target instruction-tuning data, addressing a key bottleneck to expand linguistic accessibility.
- At two model scales, SSU consistently outperforms all baselines on all core instruction-following and safety tasks. It achieves leading target-language proficiency rivaling full fine-tuning while almost perfectly preserving general source-language performance.
- Extensive analysis validates the efficacy of SSU, confirming the superiority of column-wise freezing and the importance of source data-driven parameter scoring. Qualitatively, we show that SSU avoids the linguistic code-mixing that state-of-the-art methods suffer from, explaining its superior abilities across source chat and instruction-following tasks.

2 Related Work

Language Adaptation. CPT on target language data is the standard for adapting LLMs (Cui et al., 2024; Fujii et al., 2024; Yamaguchi et al., 2024; Da Dalt et al., 2024; Cahyawijaya et al., 2024; Nguyen et al., 2024; Yamaguchi et al., 2026; Nag et al., 2025; Ji et al., 2025, *inter alia.*). While effective, it often causes catastrophic forgetting, degrading original capabilities (Tejaswi et al., 2024; Mundra et al., 2024; Yamaguchi et al., 2025). This trade-off presents a major obstacle for instruct models, where preserving core chat and instruction-following abilities is vital for their general-purpose functionality.

Catastrophic Forgetting. Mitigating catastrophic forgetting is a long-standing challenge in continual learning. Proposed solutions generally fall into five categories: (1) **Regularization** adds a penalty term to the loss function to discourage significant changes to weights deemed important for previous tasks (Kirkpatrick et al., 2017; Chen et al., 2020; Zhang et al., 2022, *inter alia.*). (2) **Replay** interleaves old and new data (de Masson d'Autume et al., 2019; Rolnick et al., 2019; Huang et al., 2024b; Sainz et al., 2025, *inter alia.*). (3) **Model merging** and post-hoc pruning mitigate forgetting by interpolating weights or removing specific task vector updates after fine-tuning (Wortsman et al., 2022; Yadav et al., 2023; Yu et al., 2024; Huang et al., 2024a, 2025, *inter alia.*). (4) **Architecture**

methods like LoRA (Hu et al., 2022) train additional new parameters while freezing the original model (Houlsby et al., 2019; Hu et al., 2022; Zhang et al., 2023, *inter alia.*). (5) **Selective parameter updates** restrict which existing weights are modified during training (Zhang et al., 2024a; Hui et al., 2025). Our work belongs to this category.

Selective Parameter Updates. While often utilized for training efficiency (Liu et al., 2021; Lodha et al., 2023; Li et al., 2023a; Pan et al., 2024; Yang et al., 2024; Li et al., 2024; Ma et al., 2024; Li et al., 2025; He et al., 2025), selective parameter updates have also proven effective for mitigating catastrophic forgetting (Zhang et al., 2024a; Hui et al., 2025). These methods can be broadly categorized as **dynamic** or **static**. Dynamic approaches alter a trainable parameter set during training, based on random selection (Li et al., 2024; Pan et al., 2024) or target data signals like gradient magnitudes (Liu et al., 2021; Li et al., 2023a; Ma et al., 2024; Li et al., 2025). In contrast, static methods define a fixed set beforehand. This allows for straightforward integration with existing pipelines, enabling the combination of orthogonal mitigation methods like regularization and replay more easily. For example, a method closest to our work (Hui et al., 2025) randomly freezes components within transformer sub-layers, while others are data-driven based on target data (Lodha et al., 2023; Zhang et al., 2024a; Panda et al., 2024; He et al., 2025).

SSU introduces a source-focused static paradigm for language adaptation. Unlike existing methods relying on random choice or target data, SSU uses a small source data sample (e.g., 500 samples) to identify and freeze parameters critical to source knowledge before adaptation. This proactively shields core abilities, offering a distinct alternative to random or target-data-driven selection criteria.

3 SSU: Selective Parameter Updates via Importance Freezing

We address adapting an instruct model using only raw, unlabeled target language data. Unlike prior work that focuses on post-hoc mitigation (Huang et al., 2024c; Yamaguchi et al., 2025), Source-Shielded Updates (SSU) targets the CPT process itself. The goal is to mitigate catastrophic forgetting during CPT, thereby maintaining the general-purpose functionality of an instruct model. Concurrently, SSU aims to achieve performance gains in the target language tasks comparable to those from full fine-tuning. Formally, given an instruct model $\mathcal{M}$, calibration data $\mathcal{D}_{\text{calib}}$, unlabeled target language data $\mathcal{D}_{\text{target}}$, and a parameter freezing ratio k, SSU adapts $\mathcal{M}$ on $\mathcal{D}_{\text{target}}$ in three stages (Figure 1).

3.1 Parameter Importance Scoring

First, SSU scores parameter importance to identify weights critical to source model capabilities. We posit that a **source-data-driven score** is suitable, as it directly aligns with the goal of preserving source knowledge. For this purpose, we adopt the importance score from Wanda (Sun et al., 2024), a popular pruning method.[2] Using a small sample of source data $\mathcal{D}_{\text{calib}}$, Wanda computes an importance score s_{ij} for each weight θ_{ij} as the product of its magnitude and the L2-norm of its corresponding input activations X_j: $s_{ij} = |\theta_{ij}| \cdot ||X_j||_2$. This identifies weights that are both large and consistently active. Scores are computed for all parameters in $\mathcal{M}$ except for the embeddings and language modeling head, as all these are updated during training following Hui et al. (2025).

3.2 Column-wise Masking

In the second stage, SSU converts element-wise importance scores into a structured freezing mask. A structured approach is crucial because naive, element-wise freezing disrupts feature transformations and causes catastrophic forgetting (Table 3). To avoid this, SSU operates at the column level. For instance, in a forward pass $Y = WX$, freezing an entire column of the weight matrix W leaves the corresponding output dimension of Y unchanged, ensuring a complete feature pathway. *The approach is analogous to protecting the core structural columns of a building during renovation; the foundational support remains untouched while peripheral elements are modified.*

Mask generation begins by aggregating scores for each column. For a weight matrix $\theta \in \mathbb{R}^{d_{out} \times d_{in}}$, a column corresponds to all parameters associated with a single input feature. The total importance score S_j for each column j is the sum of its individual scores: $S_j = \sum_i s_{ij}$. S_j robustly measures the contribution of each input feature, identifying the core structural columns to be preserved. For 1D parameters, such as biases, each

[2]While we use Wanda for its simplicity and popularity, *the SSU framework is agnostic to the importance metric.* To demonstrate this, we also evaluate two alternative source-driven scoring methods (§6).

Language	Code	Script	Family	CC Ratio
English	en	Latin	Indo-European	43.7876
Nepali	ne	Devanagari	Indo-European	.0521
Kyrgyz	ky	Cyrillic	Turkic	.0103
Amharic	am	Ge'ez	Afro-Asiatic	.0032
Hausa	ha	Latin	Afro-Asiatic	.0032
Igbo	ig	Latin	Niger-Congo	.0007

Table 1: Source (English) and target languages. Code is based on ISO 639-1, and the language-specific ratio in Common Crawl (CC Ratio) as of CC-MAIN-2025-21.

element is treated as its own column; thus, its per-weight score s_i serves as its aggregated score S_i.

The binary mask $\mathcal{B}$ for each weight matrix is generated by ranking columns by their S_j and then selecting the top $k\%$ to freeze (50% by default following Hui et al. (2025)). The corresponding columns in the mask $\mathcal{B}$ are set to 0 (freeze), while all others are set to 1 (update).

3.3 Continual Pre-training

In the third stage, the model $\mathcal{M}$ is continually pre-trained on unlabeled data $\mathcal{D}_{\text{target}}$ using a standard causal language modeling objective, denoted as the loss L. During the backward pass, the static mask $\mathcal{B}$ is applied to the gradients, zeroing out updates for frozen columns. The gradient update rule for a weight θ_{ij} is thus $\theta_{ij} \leftarrow \theta_{ij} - \eta \cdot b_{ij} \cdot \nabla_{\theta_{ij}} L$. Here, η is the learning rate, and $b_{ij} \in \{0, 1\}$ is the value from the mask $\mathcal{B}$ corresponding to the weight θ_{ij}. This method preserves knowledge stored in the most critical input-feature pathways, thus mitigating catastrophic forgetting.

4 Experimental Setup

4.1 Source Models

Following Hui et al. (2025) who used 7B and 13B models from the same family (i.e., Llama 2), we use the 7B and 13B OLMo 2 Instruct models (Walsh et al., 2025) for our experiments. The OLMo 2 models offer strong instruction-following capabilities and fully documented training data, allowing full control and transparency in our language adaptation experiments.[3]

4.2 Target Languages

We experiment with five typologically diverse languages (Table 1) that are significantly underrepresented in the training data of the source models but with wide availability of datasets with consistent task formulations (though data variations preclude direct performance comparisons between languages). These languages appear at least 840x less frequently than English in Common Crawl (CC),[4] which accounts for over 95% of the OLMo 2 pre-training corpus (Walsh et al., 2025).

4.3 Calibration and Training Data

We use tulu-3-sft-olmo-2-mixture (Lambert et al., 2025), the original instruction-tuning data for OLMo 2, for calibration (i.e., choosing which parameters to freeze). We randomly select 500 samples with a sequence length of 2,048. For CPT, we use a clean subset of MADLAD-400 (Kudugunta et al., 2023), sampling 200M tokens per language as recommended by Tejaswi et al. (2024).[5]

4.4 Baselines

We compare our approach against baselines from three categories: performance benchmarks, a reference approach from a related paradigm, and state-of-the-art methods.

Source. Off-the-shelf OLMo 2, reporting performance without any adaptation.

FFT. Full fine-tuning that updates all the parameters via CPT on target language data, quantifying the extent to which a model suffers from catastrophic forgetting without any intervention.

AdaLoRA. (Zhang et al., 2023): An architecture-based method to mitigate catastrophic forgetting. This achieves the best overall performance among LoRA-like methods in Hui et al. (2025).

HFT. A state-of-the-art **static** selective parameter update method (Hui et al., 2025). It updates 50% of parameters by randomly freezing two out of the four self-attention matrices (W_Q, W_K, W_V, W_O); and two out of three feed-forward matrices ($W_{up}, W_{down}, W_{gate}$) in a random half of the layers and one matrix in the remaining half. Since SSU is also a static method, HFT serves as a key baseline.

GMT. A state-of-the-art **dynamic** selective parameter update approach (Li et al., 2025) that drops gradients of a pre-defined ratio (50% in this

[3]While our main experiments use OLMo 2, we find that the findings generalize to OLMo 3 (see Appendix D.4).

[4]CC Ratio is from the Statistics of CC Monthly Archives.

[5]During CPT, we remove the chat template to support unlabeled data lacking role annotations (e.g., `user`).

study for fair comparison with HFT and SSU) with smaller absolute values on the target data.

To validate our use of source calibration data for scoring, we also introduce two calibration data-free ablation variants: (1) **SSU-Rand** that freezes an equal number of randomly-selected columns. This provides no principled way to preserve functionally important knowledge. (2) **SSU-Mag** that freezes columns based only on the magnitude score (i.e., $|\theta_{ij}|$; unlike $|\theta_{ij}| \cdot ||X_j||_2$ for SSU-Wanda), isolating the effect of the activation term.

We further compare SSU against other recent selective parameter update methods proposed for LLM adaptation, **LoTA** (Panda et al., 2024) and **S2FT** (Yang et al., 2024), in Appendix D.3. We find that only SSU achieves consistently both strong source preservation and high target gains.

4.5 Evaluation Benchmarks and Metrics

We report performance in the source and target languages across standard benchmarks.

Chat and Instruction-following. We report (1) **IFEval** (Zhou et al., 2023) zero-shot accuracy (strict prompt); (2) **AlpacaEval 2.0** (AE2) (Li et al., 2023b) length-controlled win-rate against GPT-4 (1106-preview) (OpenAI et al., 2024); and (3) **MT-Bench** (MTB) (Zheng et al., 2023) mean Likert-5 score over two turns; (4) **GSM8K** (Cobbe et al., 2021) five-shot exact match for multi-turn mathematical reasoning.

Safety. We use the Tülu 3 safety evaluation suite (Lambert et al., 2025, **T3**). We report the macro average score in a zero-shot setting, following Lambert et al. (2025) and Walsh et al. (2025).[6]

Source Language (English). We evaluate target-to-English machine translation (**MT**) on FLORES-200 (NLLB Team et al., 2022), reporting three-shot chrF++ (Popović, 2017) on 500 samples, following previous work (Ahia et al., 2023; Yamaguchi et al., 2025). For summarization (**SUM**) on XL-SUM (Hasan et al., 2021), we use zero-shot chrF++ on 500 samples. For machine reading comprehension (**MRC**) on Belebele (Bandarkar et al., 2024) and general reasoning on **MMLU** (Hendrycks et al., 2021), we report three-shot and five-shot accuracy, respectively, on their test sets.

[6]As instruct models typically undergo extensive safety alignment (Gemma Team et al., 2025; Lambert et al., 2025, *inter alia.*), verifying that this is not compromised during adaptation is a crucial aspect of our analysis.

Target Language. We evaluate English-to-target **MT**, **SUM**, and **MRC** on the same target-language subsets of respective datasets and settings. For reasoning, we use Global MMLU (Singh et al., 2025) and report five-shot accuracy on its test set.

We report average scores over three runs for generative tasks and use a single deterministic run with temperature zero for classification tasks. Further details (e.g., prompt templates) are in Appendix A.

5 Results

Table 2 shows performance across the four task groups: chat and instruction-following, safety, source language, and target language.

Chat and Instruction-following. Our SSU-Wanda achieves the best performance on all chat and instruction-following benchmarks, exhibiting the smallest average relative performance drops from Source of 5.9% (7B) and 4.7% (13B). This result is particularly important as these tasks directly measure core instruct model capabilities, such as multi-step reasoning and following complex constraints. The performance of SSU-Wanda demonstrates its efficacy in retaining source knowledge and abilities. The architecture-based method, AdaLoRA, performs second best with average degradations of 9.0% (7B) and 6.1% (13B). This corroborates previous findings that LoRA-style adaptation tends to forget less. However, as we discuss later, it also learn less from target data (Biderman et al., 2024; Hui et al., 2025).

In contrast, other methods exhibit more substantial performance drops. The state-of-the-art selective parameter update baselines lag considerably behind SSU-Wanda. For instance, the performance of HFT drops by 18.0% (7B) and 15.1% (13B), while the target-data-driven GMT degrades by 27.7% (7B) and 26.3% (13B). Notably, the static HFT method preserves source capabilities more effectively than the dynamic GMT method, supporting our main hypothesis that optimizing on signals from unstructured target data risks corrupting the foundational abilities of an instruct model (§1). The risk of standard adaptation is starkly illustrated by the overall performance of full fine-tuning (FFT). FFT suffers a drastic average performance loss of 34.1% (7B) and 32.3% (13B).

Finally, the low performance of baseline SSU variants (SSU-Rand and SSU-Mag) highlights the importance of the source-data-driven scoring. While both freezing random columns (SSU-Rand)

	Approach	Chat and Instruction-following (en)				Safety	Source language (en)				Target language			
		IFEval	AE2	MTB	GSM8K	T3 (en)	MT	SUM	MRC	MMLU	MT	SUM	MRC	MMLU
7B	Source	.675 +0.0	32.6 +0.0	3.98 +0.0	.796 +0.0	.851 +0.0	30.0 +0.0	22.8 +0.0	.880 +0.0	.618 +0.0	20.1 +0.0	20.2 +0.0	.334 +0.0	.304 +0.0
	FFT	.456 -32.4	10.4 -68.1	3.48 -12.5	.608 -23.6	.797 -6.4	42.8 +42.6	20.8 -8.7	.842 -4.3	.580 -6.2	30.7 +52.8	22.7 +12.4	.393 +17.7	.325 +6.8
	AdaLoRA	**.669** -0.8	24.6 -24.5	3.92 -1.5	.721 -9.4	.824 -3.2	34.1 +13.6	22.4 -1.6	.866 -1.6	.602 -2.6	19.9 -1.0	21.9 +8.4	.318 -4.8	.299 -1.8
	HFT	.621 -8.0	17.6 -45.9	3.83 -3.7	.677 -15.0	.826 -3.0	45.2 +50.6	22.3 -2.1	.854 -3.0	.595 -3.7	29.8 +48.3	22.6 +11.9	.377 +12.9	.322 +5.8
	GMT	.528 -21.7	12.5 -61.6	3.67 -7.7	.635 -20.2	.795 -6.6	45.5 +51.6	21.6 -5.1	.841 -4.4	.582 -5.8	30.9 +53.8	**22.9** +13.4	.385 +15.3	.319 +4.8
	SSU-Rand	.608 -9.9	18.0 -44.7	3.81 -4.2	.683 -14.2	.835 -1.9	45.5 +51.6	22.4 -1.6	.861 -2.2	.597 -3.4	30.2 +50.3	22.7 +12.4	.394 +18.0	.324 +6.4
	SSU-Mag	.570 -15.5	14.9 -54.2	3.78 -5.0	.655 -17.7	.822 -3.4	44.7 +48.9	22.0 -3.4	.859 -2.4	.593 -4.1	29.7 +47.8	22.7 +12.4	.383 +14.7	.319 +4.8
	SSU-Wanda	**.669** -0.8	**27.0** -17.1	**3.96** -0.5	**.752** -5.5	**.850** -0.1	**45.7** +52.3	**22.8** +0.1	**.869** -1.3	**.606** -2.0	**31.0** +54.3	22.8 +12.9	**.403** +20.7	**.333** +9.4
13B	Source	.763 +0.0	37.2 +0.0	4.06 +0.0	.853 +0.0	.821 +0.0	33.3 +0.0	24.5 +0.0	.897 +0.0	.665 +0.0	22.4 +0.0	20.7 +0.0	.374 +0.0	.329 +0.0
	FFT	.448 -41.3	14.5 -61.1	3.52 -13.3	.740 -13.3	.737 -10.2	40.1 +20.3	15.7 -35.8	.892 -0.5	.647 -2.7	33.6 +50.1	22.9 +10.4	**.492** +31.6	**.361** +9.8
	AdaLoRA	.719 -5.8	32.1 -13.8	**4.05** -0.2	.815 -4.5	.799 -2.7	36.6 +9.8	**24.4** -0.2	**.898** +0.1	.660 -0.8	23.0 +2.7	22.3 +7.5	.365 -2.4	.311 -5.4
	HFT	.631 -17.3	25.8 -30.7	3.92 -3.4	.776 -9.0	.785 -4.4	44.1 +32.2	20.7 -15.3	.894 -0.3	.658 -1.1	33.7 +50.5	22.8 +9.9	.476 +27.3	.355 +8.0
	GMT	.497 -34.9	19.3 -48.2	3.64 -10.3	.754 -11.6	.755 -8.0	37.5 +12.5	16.5 -32.5	.896 -0.1	.654 -1.7	33.5 +49.6	22.8 +9.9	.473 +26.5	.353 +7.4
	SSU-Rand	.630 -17.5	24.7 -33.7	3.89 -4.1	.781 -8.5	.783 -4.6	43.9 +31.6	21.7 -11.3	**.898** +0.1	.656 -1.4	33.6 +50.1	23.0 +10.9	.478 +27.8	.356 +8.3
	SSU-Mag	.572 -25.1	20.6 -44.7	3.80 -6.4	.763 -10.6	.776 -5.5	40.2 +20.6	20.2 -17.4	.892 -0.5	.657 -1.2	32.8 +46.5	22.6 +8.9	.467 +24.9	.350 +6.5
	SSU-Wanda	**.730** -4.4	**33.4** -10.3	**4.05** -0.2	**.822** -3.7	**.805** -2.0	**48.2** +44.5	24.2 -1.0	.897 +0.0	**.661** -0.6	**34.1** +52.3	**23.2** +11.8	.486 +29.9	.359 +9.2

Table 2: Aggregated average performance across all languages per task. Green denotes scores better than Source with subscripts showing relative changes (%). **Bold** and underlined indicate best and second-best methods for each model scale. Tables 9, 10, 11, and 12 include a full suite of results.

and columns selected by magnitude alone (SSU-Mag) outperform FFT, they substantially underperform SSU-Wanda. SSU-Rand performance is 18.2% (7B) and 16.0% (13B) lower than Source, while SSU-Mag causes even greater drops of 23.0% (7B) and 21.7% (13B). The substantial underperformance of these calibration data-free approaches underscores the critical need for a source-data-informed importance scoring method for preserving the core capabilities of an instruct model in the source language. As we demonstrate in §6, this principle is not limited to Wanda; other source-data-driven scoring methods are also highly effective, confirming the versatility of the SSU framework.

Safety. SSU-Wanda also best preserves the safety alignment of the source, with small performance drops of only 0.1% (7B) and 2.0% (13B) compared to Source. In contrast, FFT and the target-data-driven GMT cause large drops, with safety scores dropping by up to 10.2%. While other selective methods partially preserve source performance, they still lag behind SSU-Wanda.

Source Language. SSU-Wanda not only preserves source capabilities but also enhances them in the cross-lingual translation task. It ranks top for the 7B model across all benchmarks and leads in MT and MMLU for the 13B model with a close second in SUM and MRC. Notably, its performance on target-to-English MT improves substantially by up to 52.3% relative to Source. Monolingual task performance (SUM, MRC, and MMLU) is almost perfectly maintained, with relative drops never exceeding 2.0% (7B) and 1.0% (13B). AdaLoRA is the second-best performer overall, also showing strong preservation across monolingual tasks. However, its gains in the MT task are substantially smaller, the worst among all approaches. This suggests that while LoRA-based methods effectively prevent forgetting, the structural isolation of their updates may be less adept at integrating new linguistic knowledge for complex cross-lingual tasks. The remaining adaptation methods generally exhibit greater performance degradation than SSU-Wanda, consistent with instruction-following and safety results.

Target Language. Finally, SSU-Wanda demonstrates exceptional performance on target language tasks, securing the best results across all benchmarks for both model scales in the majority of cases. Crucially, its performance is highly competitive with FFT, even surpassing it on all benchmarks for 7B models and on half for 13B models. The performance difference between SSU and FFT is consistently minimal, confirming that SSU-Wanda achieves the target-language gains of a full update with drastically smaller catastrophic forgetting. This aligns with observations from optimization theory, arguing that freezing parameters acts as a regularization term that stabilizes training and en-

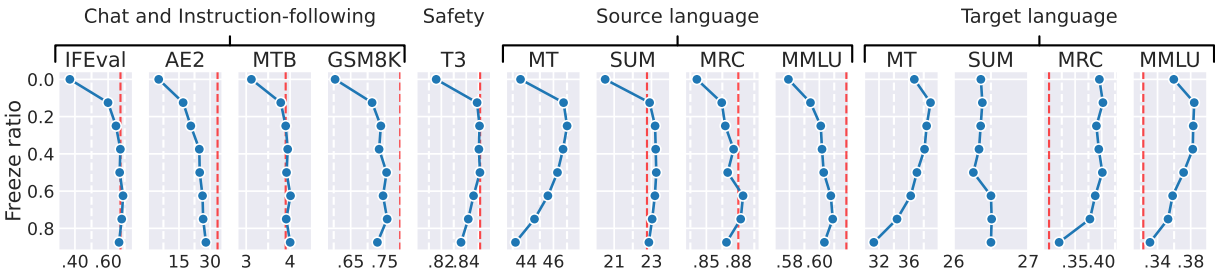

Figure 2: Performance across freezing ratios using SSU on Igbo as target language. The dashed red line indicates Source performance (omitted for MT and SUM due to very low scores). See §4.5 for details on evaluation metrics.

ables a sparse fine-tuned model to match or exceed the performance of its dense counterpart (Fu et al., 2023; Zhang et al., 2024b; Hui et al., 2025). All the other selective parameter update methods also yield solid improvements, though typically smaller than those of SSU-Wanda. In contrast, AdaLoRA shows the smallest improvement and often fails to surpass the source model. This confirms that LoRA-based methods have a smaller inductive bias from the target data (Biderman et al., 2024; Hui et al., 2025). This highlights the unique effectiveness of SSU-Wanda, which successfully masters tasks in the target language while preserving its original knowledge and abilities in the source.

Overall, SSU-Wanda demonstrates the benefits of full fine-tuning without the associated catastrophic forgetting, consistently outperforming all other evaluated methods.

6 Analysis

This section evaluates the robustness of the SSU framework by isolating the impact of core design choices and hyperparameters. Due to resource constraints, we use the 7B model with our primary method, SSU-Wanda. We select Igbo as the target language, as it is the most underrepresented language among our target languages (Table 1).

Parameter Freezing Ratio. While we use a default 50% freezing ratio for fair comparison with baselines following Hui et al. (2025), this hyperparameter impacts performance. We therefore evaluate freezing ratios from 0% (defaulting to FFT) to 87.5% in 12.5% increments. Figure 2 shows that source language performance, such as chat and safety, generally improves with higher freezing ratios. In contrast, performance on target language tasks often shows an opposite trend, degrading as more parameters are frozen, with a particularly sharp drop in MMLU after reaching a 37.5% ratio. Target-to-English MT is a notable exception. Although the models generate English text, performance declines as the freezing ratio increases, particularly after 37.5%. This trend contradicts other source tasks. This occurs because MT requires knowledge of both source and target languages.

Our results show a trade-off between source knowledge retention and target language acquisition. Therefore, we recommend practitioners tailor the freezing ratio to specific goals: **General purpose**: A default 50% ratio offers balanced performance. **Source-capability priority**: A higher ratio ($\geq 60\%$) is optimal, as performance on tasks like IFEval, MRC, and MMLU plateaus around this point. **Target-language priority**: A lower ratio ($\leq 40\%$) is preferable, given the performance drops observed in MT and MMLU beyond this threshold. We extend this analysis to baselines in Appendix D.1, finding that HFT consistently underperforms SSU despite following a similar performance-scaling pattern, while GMT fails to preserve source capabilities regardless of the ratio.

Alternative Freezing Methods. SSU employs column-wise freezing to preserve the entire processing pathway of critical source features (§3.2). To validate this design choice, we compare its effectiveness against row-wise and element-wise freezing. As shown in Table 3 ①, the results demonstrate a clear advantage for our column-wise approach. Column-wise freezing consistently achieves the best performance on chat, safety, and source language tasks.[7] On target language tasks, it remains highly competitive, with only a 1.2 point drop on MT compared to element-wise freezing. These results validate the guiding hypothesis for the design of SSU: *preserving entire feature pathways is a critical strategy to safeguard source knowledge while*

[7]While row-wise freezing preserves all connections from a single input neuron, it fails to protect any single, complete output feature. This explains its weaker performance across chat, safety, and source language tasks.

	Approach	Chat and Instruction-following IFEval	AE2	MTB	GSM8K	Safety T3	Source language MT	SUM	MRC	MMLU	Target language (Igbo) MT	SUM	MRC	MMLU
	Source	.675	32.6	3.98	.796	.851	28.5	22.8	.880	.618	23.0	23.3	.301	.323
	SSU (Default)	.670	25.0	3.92	.756	.851	46.3	23.3	.870	.603	37.1	26.3	.401	.371
①	Row-wise	.548	11.3	3.74	.675	.846	46.0	21.8	.862	.598	36.9	26.5	.407	.358
①	Element-wise	.457	7.7	3.35	.657	.829	46.4	21.1	.851	.587	38.3	26.5	.399	.370
②	SSU-Rand	.564	12.5	3.75	.680	.838	45.9	22.4	.856	.597	37.3	26.4	.401	.355
②	SSU-Mag	.497	8.9	3.59	.638	.828	45.1	21.7	.852	.592	36.6	26.2	.379	.348
②	SSU-SparseGPT	.678	24.5	3.89	.751	.843	46.2	23.1	.876	.604	37.2	26.5	.400	.372
②	SSU-FIM	.669	26.3	3.94	.747	.847	46.4	23.2	.874	.609	37.1	26.5	.399	.371
③	Alpaca	.673	24.0	3.97	.750	.849	46.7	23.1	.874	.604	37.1	26.2	.394	.379

Table 3: Ablation studies on ① freezing structures, ② importance scoring methods, and ③ calibration data sources. All models are evaluated on 7B and Igbo as the target language. "Default" denotes SSU-Wanda with column-wise freezing using the original tulu-3-sft-olmo-2-mixture data for calibration.

enabling effective target-language adaptation. We provide a theoretical grounding for these structural constraints and their relation to the stability-plasticity dilemma in Appendix D.5.

Alternative Importance Scoring Methods. SSU is compatible with importance scoring methods beyond Wanda. To demonstrate this, we evaluate two source-data-driven methods: SparseGPT (Frantar and Alistarh, 2023) and the diagonal of the Fisher Information Matrix (Kirkpatrick et al., 2017, FIM); see Appendix B for details. In monolingual source tasks, SSU-SparseGPT and SSU-FIM show comparable average performance drops (4.3% and 3.5%, respectively) to SSU-Wanda (4.0%), as shown in Table 3 ②. This contrasts sharply with the larger drops of data-free variants like SSU-Rand (13.5%) and SSU-Mag (17.9%). These findings demonstrate the versatility of SSU, offering strong performance across various source-data-driven scoring methods.

Calibration Data for Parameter Importance Scoring. SSU-Wanda requires source calibration data to identify critical model weights since it relies on Wanda for parameter importance scoring. While we use the original instruction-tuning data for OLMo 2 in our main experiments, this is often unavailable for other frontier models. We therefore investigate the efficacy of using an alternative, publicly available dataset. Specifically, we use Alpaca (Taori et al., 2023) as the calibration dataset and follow the same preprocessing and training procedures as the original data. Table 3 ③ shows that performance with Alpaca is highly comparable to that with the original data, with a maximum difference of 1.0, demonstrating the robustness of SSU-Wanda to the choice of calibration data. We observe similar robustness regarding calibration data size; reducing samples from 500 to 128 yields negligible performance differences (see Appendix D.2).

	Approach	HumanEval (↑) ne	ky	am	ha	ig
7B	Source	.445 $_{+0.0}$	.445 $_{+0.0}$	.445 $_{+0.0}$	.445 $_{+0.0}$	.445 $_{+0.0}$
7B	FFT	.287 $_{-35.5}$	.226 $_{-49.2}$	.268 $_{-39.8}$	.128 $_{-71.2}$	.201 $_{-54.8}$
7B	AdaLoRA	**.451** $_{+1.3}$	**.384** $_{-13.7}$	.354 $_{-20.5}$	.323 $_{-27.4}$	.348 $_{-21.8}$
7B	HFT	.384 $_{-13.7}$	.335 $_{-24.7}$	.366 $_{-17.8}$	.323 $_{-27.4}$	.354 $_{-20.5}$
7B	GMT	.274 $_{-38.4}$	.287 $_{-35.5}$	.323 $_{-27.4}$	.177 $_{-60.2}$	.256 $_{-42.5}$
7B	SSU-Rand	.396 $_{-11.0}$	.335 $_{-24.7}$	.323 $_{-27.4}$	.262 $_{-41.1}$	.323 $_{-27.4}$
7B	SSU-Mag	.317 $_{-28.8}$	.293 $_{-34.2}$	.311 $_{-30.1}$	.268 $_{-39.8}$	.293 $_{-34.2}$
7B	SSU-Wanda	.402 $_{-9.7}$	**.384** $_{-13.7}$	**.396** $_{-11.0}$	**.390** $_{-12.4}$	**.421** $_{-5.4}$
13B	Source	.524 $_{+0.0}$	.524 $_{+0.0}$	.524 $_{+0.0}$	.524 $_{+0.0}$	.524 $_{+0.0}$
13B	FFT	.445 $_{-15.1}$	.317 $_{-39.5}$	.451 $_{-14.0}$	.152 $_{-71.0}$	.152 $_{-71.0}$
13B	AdaLoRA	.476 $_{-9.2}$	.457 $_{-12.9}$	.500 $_{-4.7}$	.433 $_{-17.4}$	.476 $_{-9.2}$
13B	HFT	.451 $_{-14.0}$	.439 $_{-16.3}$	.451 $_{-14.0}$	.378 $_{-27.9}$	.433 $_{-17.4}$
13B	GMT	.451 $_{-14.0}$	.439 $_{-16.3}$	.463 $_{-11.7}$	.360 $_{-31.3}$	.378 $_{-27.9}$
13B	SSU-Rand	**.524** $_{-0.1}$	.451 $_{-14.0}$	.463 $_{-11.7}$	.451 $_{-14.0}$	.415 $_{-20.9}$
13B	SSU-Mag	.415 $_{-20.9}$	.427 $_{-18.6}$	.427 $_{-18.6}$	.250 $_{-52.3}$	.329 $_{-37.3}$
13B	SSU-Wanda	.482 $_{-8.1}$	**.512** $_{-2.4}$	**.537** $_{+2.4}$	**.482** $_{-8.1}$	**.500** $_{-4.7}$

Table 4: Coding performance on HumanEval (pass@1). **Bold** and underlined indicate best and second-best methods for each model scale.

Universality of Shielded Parameters. We investigate whether shielded parameters are specific to the English language. We hypothesize that SSU preserves universal functional units, such as logic and reasoning, rather than surface-level linguistic features. To evaluate this, we measure performance on HumanEval (Chen et al., 2021), where logic transcends natural language. Table 4 demonstrates that SSU-Wanda maintains coding proficiency near the levels of Source. In contrast, FFT and GMT suffer substantial degradation. For the 7B models, SSU-Wanda shows a 10.4% average relative performance drop, whereas FFT suffers a severe loss of 49.7%. The 13B models exhibit a comparable

Approach	Chat and Instruction-following				Safety	Source language				Target language (Igbo)			
	IFEval	AE2	MTB	GSM8K	T3	MT	SUM	MRC	MMLU	MT	SUM	MRC	MMLU
Source	.675	32.6	3.98	.796	.851	28.5	22.8	.880	.618	23.0	23.3	.301	.323
FFT	.645	17.1	3.95	.685	.835	42.6	21.9	.857	.604	**30.9**	26.2	.341	.349
AdaLoRA	.678	**30.6**	**4.05**	.750	.837	28.4	22.5	.874	**.614**	16.7	24.8	.270	.318
HFT	**.693**	25.1	3.89	.732	.841	42.3	22.4	.870	.607	29.3	**26.6**	.328	.346
GMT	.665	23.2	3.93	.726	.838	**43.0**	22.5	**.879**	.611	30.7	26.3	.349	.347
SSU-Rand	.682	24.4	3.95	.729	.831	42.0	22.5	.871	.610	28.9	26.3	.337	.343
SSU-Mag	.664	21.3	3.97	.704	.831	42.6	22.2	.874	.607	28.9	26.2	.340	.336
SSU-Wanda	.671	27.9	3.98	**.783**	**.848**	42.3	**22.6**	.878	.613	28.9	**26.6**	**.357**	**.352**

Table 5: Performance of models adapted with 20M tokens of target language data. All models are evaluated on 7B and Igbo as the target language. **Bold** and underlined indicate best and second-best methods.

trend, with SSU-Wanda declining by only 4.2%. These results confirm that SSU safeguards fundamental capabilities, such as reasoning and logic, which are shared across languages. A proxy analysis regarding target-language instruction-following in Appendix D.6 further supports these findings.

Ultra-low-resource Settings. To evaluate the efficacy of SSU under extreme data constraints, we adapt models using only 20M tokens, representing 10% of our default adaptation set. As shown in Table 5, SSU-Wanda achieves the best or second best performances in 10 out of 13 tasks in this ultra-low-resource regime. While the reduced training data naturally limits overall weight drift, SSU-Wanda exhibits substantially better retention of core capabilities (AE2, GSM8K, Safety) than baselines, which show immediate degradation even with minimal updates. AdaLoRA remains a notable exception, as it "learns less and forgets less" (Biderman et al., 2024; Hui et al., 2025), resulting in strong source retention but substantially weaker target-language acquisition. Furthermore, SSU-Wanda achieves target-language improvements in SUM (26.6), MRC (.357), and MMLU (.352) that exceed those of FFT. This confirms that shielding critical source parameters acts as a beneficial regularizer for acquiring target linguistic features even when training data is scarce.

Qualitative Analysis. SSU-Wanda surpasses other state-of-the-art selective parameter update baselines across all chat and instruction-following benchmarks (Table 2). This performance gap stems partly from the susceptibility of baseline methods to code-mixing (i.e., the unintentional blending of multiple languages in responses) or generating responses entirely in the target language, despite English instructions. Specifically, analyzing the language ratio in generated responses on AE2 shows that SSU restricts code-mixing to merely 1.0% of its responses on average for the 7B models. In contrast, HFT and GMT generate code-mixed text in 6.4% and 16.9%, respectively.[8] This substantial reduction in the occurrence of code-mixing reflects the more robust retention of source language abilities and superior chat performance. A typical example of this behavior for models trained on Igbo is provided below.

Instruction in EN: How do I take care of a wooden table?

HFT Response: *To take care nke a wood table, clean ya na a soft duster ma ọ bụ microfiber towel iji wepụ dust na grime. N'ihe banyere stains, mix otu teaspoon nke mild detergent na mmiri na-a teaspoon...* (*continued.*)

SSU Response: *To take care, clean your wooden table regularly with mild soap and water. Use a soft cloth for polishing, applying wood polish or beeswax to maintain its natural finish. Avoid placing hot items directly on the surface to prevent scratches. Keep it away from direct sunlight and excessive moisture.*

7 Conclusion

We introduced **S**ource-**S**hielded **U**pdates (**SSU**) for language adaptation of instruct models using only unlabeled target language data. Our SSU framework proactively identifies critical source knowledge using an importance scoring method and a small set of source calibration data. It then shields this knowledge via a column-wise freezing strategy before adaptation, effectively preventing catastrophic forgetting in the source language. Extensive experiments across five languages and two model scales show that SSU best preserves crucial source capabilities, such as instruction-following and safety, over strong baselines while achieving target language proficiency matching or surpassing full fine-tuning. This work provides an effective and scalable pathway to expand the linguistic reach of instruct models without costly, specialized data, opening avenues for robust model adaptation.

[8] We use GlotLID (Kargaran et al., 2023, Commit 28d4264) to identify code-mixed responses where normalized English confidence falls below 0.9.

Limitations

Baselines Scope. This paper primarily compares SSU against state-of-the-art selective parameter update methods for LLM adaptation, specifically HFT and GMT. Additional evaluations against LoTA and S2FT are provided in Appendix D.3 to ensure an extensive evaluation. Strategies such as source data mixing (Zheng et al., 2024; Sainz et al., 2025) and model merging and post-hoc pruning (Blevins et al., 2024; Huang et al., 2025) are orthogonal to this work (as discussed in §2). Furthermore, foundational continual learning methods for task-incremental learning, such as HAT (Serra et al., 2018), remain computationally prohibitive for billion-parameter models (see Appendix E for discussion). Consequently, this work prioritizes scalable, LLM-specific methods for comparison to maintain practical relevance. Exploring the synergy between SSU and orthogonal strategies such as model merging or replay remains a promising direction for future research.

Hyperparameter Selection. Due to the substantial computational cost of fine-tuning and evaluating 100+ adapted models (e.g., Table 2 encompasses 70 adapted models), this study does not perform exhaustive hyperparameter searches for all approaches including both baselines and the proposed method. Instead, the experimental protocol follows established language adaptation literature for instruct models (Yamaguchi et al., 2025). For freezing ratios, this work adopts the 50% sparsity level used in HFT (Hui et al., 2025) to facilitate fair comparison, with sensitivity analysis provided in §6 and Appendix D.1. While reported performance might not represent the global optimum for each method across languages, avoiding exhaustive tuning prevents introducing bias toward methods with larger search spaces. Utilizing a standard configuration ensures a rigorous and equitable evaluation of the underlying methods.

Ethical Considerations

While the current study on SSU should not present immediate ethical conflicts given its scope on catastrophic forgetting mitigation, the deployment of adapted instruct models in underrepresented languages (e.g., Nepali, Kyrgyz, or Amharic) requires further scrutiny. These adapted models may unintentionally reinforce harmful biases or introduce safety vulnerabilities that standard benchmarks fail to detect. Consequently, responsible deployment and continued research into cross-lingual safety alignment remain essential.

Acknowledgements

We would like to thank Mingzi Cao, Xingwei Tan, and Huiyin Xue for their valuable feedback. We acknowledge (1) IT Services at the University of Sheffield for the provision of services for high-performance computing; (2) the use of the University of Oxford Advanced Research Computing (ARC) facility; and (3) EuroHPC Joint Undertaking for awarding us access to MeluXina at LuxProvide, Luxembourg. AY is supported by the Engineering and Physical Sciences Research Council (EPSRC) [grant number EP/W524360/1] and the Japan Student Services Organization (JASSO) Student Exchange Support Program (Graduate Scholarship for Degree Seeking Students). AV research is partly supported by UKRI (grants MR/U506734/1 and EP/T02450X/1), CNPq (406926/2025-5) and EQUATE. NA is partly supported by AstraZeneca and the EPSRC (grant EP/Y009800/1).

References

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore. Association for Computational Linguistics.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III*, page 144–161, Berlin, Heidelberg. Springer-Verlag.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, and 30 others. 2024. PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, page 929–947, New York, NY, USA. Association for Computing Machinery.

Israel Abebe Azime, Atnafu Lambebo Tonja, Tadesse Destaw Belay, Mitiku Yohannes Fuge, Aman Kassahun Wassie, Eyasu Shiferaw Jada, Yonas Chanie, Walelign Tewabe Sewunetie, and Seid Muhie Yimam. 2024. Walia-LLM: Enhancing Amharic-LLaMA by integrating task-specific and generative datasets. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 432–444, Miami, Florida, USA. 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.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. 2024. LoRA learns less and forgets less. *Transactions on Machine Learning Research*. Featured Certification.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Terra Blevins, Tomasz Limisiewicz, Suchin Gururangan, Margaret Li, Hila Gonen, Noah A. Smith, and Luke Zettlemoyer. 2024. Breaking the curse of multilinguality with cross-lingual expert language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10822–10837, Miami, Florida, USA. Association for Computational Linguistics.

Samuel Cahyawijaya, Holy Lovenia, Fajri Koto, Rifki Putri, Wawan Cenggoro, Jhonson Lee, Salsabil Akbar, Emmanuel Dave, Nuurshadieq Nuurshadieq, Muhammad Mahendra, Rr Putri, Bryan Wilie, Genta Winata, Alham Aji, Ayu Purwarianti, and Pascale Fung. 2024. Cendol: Open instruction-tuned generative large language models for Indonesian languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14899–14914, Bangkok, Thailand. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *arXiv preprint*, arXiv:2107.03374.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7870–7881, Online. 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.

Yiming Cui, Ziqing Yang, and Xin Yao. 2024. Efficient and effective text encoding for Chinese LLaMA and Alpaca. *arXiv preprint*, arXiv:2304.08177v3.

Severino Da Dalt, Joan Llop, Irene Baucells, Marc Pamies, Yishi Xu, Aitor Gonzalez-Agirre, and Marta Villegas. 2024. FLOR: On the effectiveness of language adaptation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7377–7388, Torino, Italia. ELRA and ICCL.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *Proceedings of the Twelfth International Conference on Learning Representations*.

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

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

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):12799–12807.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual LLM adaptation: Enhancing Japanese language capabilities. In *Proceedings of the First Conference on Language Modeling*.

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. 2023. A framework for few-shot language model evaluation. https://zenodo.org/records/10256836.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *arXiv preprint*, arXiv:2503.19786.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2015. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint*, arXiv:1312.6211v3.

Stephen Grossberg. 1982. *Studies of mind and brain: neural principles of learning, perception, development, cognition, and motor control*. Boston studies in the philosophy of science; 70. D. Reidel Publishing Company.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. LightEval: A lightweight framework for LLM evaluation. GitHub repository.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Haoze He, Juncheng B Li, Xuan Jiang, and Heather Miller. 2025. SMT: Fine-tuning large language models with sparse matrices. In *Proceedings of the Thirteenth International Conference on Learning Representations*.

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

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 *Proceedings of the Tenth International Conference on Learning Representations*.

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. 2024a. EMR-Merging: Tuning-free high-performance model merging. In *Advances in Neural Information Processing Systems*, volume 37, pages 122741–122769. Curran Associates, Inc.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12365–12394, Singapore. Association for Computational Linguistics.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024b. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1416–1428, Bangkok, Thailand. Association for Computational Linguistics.

Shih-Cheng Huang, Pin-Zu Li, Yu-chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tsai, and Hung-yi Lee. 2024c. Chat vector: A simple approach to equip LLMs with instruction following and model alignment in new languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10943–10959, Bangkok, Thailand. Association for Computational Linguistics.

Wei Huang, Anda Cheng, and Yinggui Wang. 2025. Mitigating catastrophic forgetting in large language models with forgetting-aware pruning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21853–21867, Suzhou, China. Association for Computational Linguistics.

Tingfeng Hui, Zhenyu Zhang, Shuohuan Wang, Weiran Xu, Yu Sun, and Hua Wu. 2025. HFT: Half fine-tuning for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12791–12819, Vienna, Austria. Association for Computational Linguistics.

Shaoxiong Ji, Zihao Li, Indraneil Paul, Jaakko Paavola, Peiqin Lin, Pinzhen Chen, Dayyán O'Brien, Hengyu Luo, Hinrich Schütze, Jörg Tiedemann, and Barry Haddow. 2025. EMMA-500: Enhancing massively multilingual adaptation of large language models. *arXiv preprint*, arXiv:2409.17892v3.

Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schuetze. 2023. GlotLID: Language identification for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6155–6218, Singapore. Association for Computational Linguistics.

Zixuan Ke, Bing Liu, and Xingchang Huang. 2020. Continual learning of a mixed sequence of similar and dissimilar tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 18493–18504. Curran Associates, Inc.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Tatsuya Konishi, Mori Kurokawa, Chihiro Ono, Zixuan Ke, Gyuhak Kim, and Bing Liu. 2023. Parameter-level soft-masking for continual learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17492–17505. PMLR.

Sneha Kudugunta, Isaac Rayburn Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. MADLAD-400: A multilingual and document-level large audited dataset. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025. Tulu 3: Pushing frontiers in open language model post-training. In *Proceedings of the Second Conference on Language Modeling*.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, and 13 others. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hanqi Li, Lu Chen, Da Ma, Zijian Wu, Su Zhu, and Kai Yu. 2024. Evolving subnetwork training for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 27547–27562. PMLR.

Haoling Li, Xin Zhang, Xiao Liu, Yeyun Gong, Yifan Wang, Qi Chen, and Peng Cheng. 2025. Enhancing large language model performance with gradient-based parameter selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24431–24439.

Sheng Li, Geng Yuan, Yue Dai, Youtao Zhang, Yanzhi Wang, and Xulong Tang. 2023a. SmartFRZ: An efficient training framework using attention-based layer freezing. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. AlpacaEval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`. GitHub repository.

Yuhan Liu, Saurabh Agarwal, and Shivaram Venkataraman. 2021. AutoFreeze: Automatically freezing model blocks to accelerate fine-tuning. *arXiv preprint*, arXiv:2102.01386.

Abhilasha Lodha, Gayatri Belapurkar, Saloni Chalkapurkar, Yuanming Tao, Reshmi Ghosh, Samyadeep Basu, Dmitrii Petrov, and Soundararajan Srinivasan. 2023. On surgical fine-tuning for language encoders. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3105–3113, Singapore. Association for Computational Linguistics.

Da Ma, Lu Chen, Pengyu Wang, Hongshen Xu, Hanqi Li, Liangtai Sun, Su Zhu, Shuai Fan, and Kai Yu. 2024. Sparsity-accelerated training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14696–14707, Bangkok, Thailand. Association for Computational Linguistics.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, page 72–88, Berlin, Heidelberg. Springer-Verlag.

Arun Mallya and Svetlana Lazebnik. 2018. PackNet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7765–7773.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft. GitHub repository.

Nandini Mundra, Aditya Nanda Kishore Khandavally, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, and Mitesh M Khapra. 2024. An empirical comparison of vocabulary expansion and initialization approaches for language models. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 84–104, Miami, FL, USA. Association for Computational Linguistics.

Arijit Nag, Soumen Chakrabarti, Animesh Mukherjee, and Niloy Ganguly. 2025. Efficient continual pre-training of LLMs for low-resource languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 304–317, Albuquerque, New Mexico. Association for Computational Linguistics.

Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Zhiqiang Hu, Chenhui Shen, Yew Ken Chia, Xingxuan Li, Jianyu Wang, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, Hang Zhang, and Lidong Bing. 2024. SeaLLMs - large language models for Southeast Asia. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 294–304, Bangkok, Thailand. Association for Computational Linguistics.

NLLB Team, 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. *arXiv preprint*, arXiv:2207.04672.

OpenAI. 2025. GPT-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf. Technical report.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. GPT-4 technical report. *arXiv preprint*, arXiv:2303.08774v6.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. 2024. Lisa: Layerwise importance sampling for memory-efficient large language model fine-tuning. In *Advances in Neural Information Processing Systems*, volume 37, pages 57018–57049. Curran Associates, Inc.

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

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Oscar Sainz, Naiara Perez, Julen Etxaniz, Joseba Fernandez de Landa, Itziar Aldabe, Iker García-Ferrero, Aimar Zabala, Ekhi Azurmendi, German Rigau, Eneko Agirre, Mikel Artetxe, and Aitor Soroa. 2025. Instructing large language models for low-resource languages: A systematic study for Basque. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29124–29148, Suzhou, China. Association for Computational Linguistics.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4548–4557. PMLR.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the Third International Conference on Learning Representations*, pages 1–14.

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.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *Proceedings of the Twelfth*

International Conference on Learning Representations.

Mingxu Tao, Chen Zhang, Quzhe Huang, Tianyao Ma, Songfang Huang, Dongyan Zhao, and Yansong Feng. 2024. Unlocking the potential of model merging for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8705–8720, Miami, Florida, USA. Association for Computational Linguistics.

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

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, and 49 others. 2025. Olmo 3. *arXiv preprint*, arXiv:2512.13961.

Atula Tejaswi, Nilesh Gupta, and Eunsol Choi. 2024. Exploring design choices for building language-specific LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10485–10500, Miami, Florida, USA. Association for Computational Linguistics.

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.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, and 23 others. 2025. 2 OLMo 2 furious (COLM's version). In *Proceedings of the Second Conference on Language Modeling*.

Wenjin Wang, Yunqing Hu, Qianglong Chen, and Yin Zhang. 2023. Task difficulty aware parameter allocation & regularization for lifelong learning. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7776–7785.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *Proceedings of the Tenth International Conference on Learning Representations*.

Miles Williams and Nikolaos Aletras. 2024. On the impact of calibration data in post-training quantization and pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10100–10118, Bangkok, Thailand. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. 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 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

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*, volume 36, pages 7093–7115. Curran Associates, Inc.

Atsuki Yamaguchi, Terufumi Morishita, Aline Villavicencio, and Nikolaos Aletras. 2025. Adapting chat language models using only target unlabeled language data. *Transactions on Machine Learning Research*.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. 2024. An empirical study on cross-lingual vocabulary adaptation for efficient language model inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6760–6785, Miami, Florida, USA. Association for Computational Linguistics.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. 2026. How can we effectively expand the vocabulary of LLMs with 0.01GB of target language text? *Computational Linguistics*, 52(1):295–330.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint*, arXiv:2505.09388.

Xinyu Yang, Jixuan Leng, Geyang Guo, Jiawei Zhao, Ryumei Nakada, Linjun Zhang, Huaxiu Yao, and Beidi Chen. 2024. S^2FT: Efficient, scalable and generalizable LLM fine-tuning by structured sparsity. In *Advances in Neural Information Processing Systems*, volume 37, pages 59912–59947. Curran Associates, Inc.

Zheng Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, Genta Winata, Stella Biderman, Edward Raff, Dragomir Radev, and Vassilina Nikoulina. 2023. BLOOM+1: Adding language support to BLOOM for zero-shot prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11682–11703, Toronto, Canada. Association for Computational Linguistics.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57755–57775. PMLR.

Bo Zeng, Chenyang Lyu, Sinuo Liu, Mingyan Zeng, Minghao Wu, Xuanfan Ni, Tianqi Shi, Yu Zhao, Yefeng Liu, Chenyu Zhu, Ruizhe Li, Jiahui Geng, Qing Li, Yu Tong, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025. Marco-bench-MIF: On multilingual instruction-following capability of large language. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24058–24072, Vienna, Austria. Association for Computational Linguistics.

Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR.

Han Zhang, Sheng Zhang, Yang Xiang, Bin Liang, Jinsong Su, Zhongjian Miao, Hui Wang, and Ruifeng Xu. 2022. CLLE: A benchmark for continual language learning evaluation in multilingual machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 428–443, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hengyuan Zhang, Yanru Wu, Dawei Li, Sak Yang, Rui Zhao, Yong Jiang, and Fei Tan. 2024a. Balancing speciality and versatility: a coarse to fine framework for supervised fine-tuning large language model. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7467–7509, Bangkok, Thailand. Association for Computational Linguistics.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Zhi Zhang, Qizhe Zhang, Zijun Gao, Renrui Zhang, Ekaterina Shutova, Shiji Zhou, and Shanghang Zhang. 2024b. Gradient-based parameter selection for efficient fine-tuning. In *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28566–28577.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. 2025. Spurious forgetting in continual learning of language models. In *Proceedings of the Thirteenth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Wenzhen Zheng, Wenbo Pan, Xu Xu, Libo Qin, Li Yue, and Ming Zhou. 2024. Breaking language barriers: Cross-lingual continual pre-training at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7725–7738, Miami, Florida, USA. Association for Computational Linguistics.

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

Appendix Directory

- **Appendix A:** Evaluation Details
- **Appendix B:** Implementation Details
  - General Setup
  - Alternative Scoring Method Implementations
- **Appendix C:** Supplementary Results
- **Appendix D:** Supplementary Analysis
  - Impact of Freezing Ratio on Baselines
  - Calibration Data Size for Parameter Importance Scoring
  - Comparison to Additional Baselines
  - Generalization to OLMo 3 Architecture
  - Theoretical Analysis
  - Proxy Evaluation on Target-Language Instruction-following
- **Appendix E:** Extended Related Work
- **Appendix F:** License
- **Appendix G:** Use of Generative AI Tools

A Evaluation Details

LLM-as-a-Judge. Following Yamaguchi et al. (2025), we use judgments from GPT-4.1 nano (2025-04-14) for AE2 and `Flow-Judge-v0.1` for MTB.

Prompt Templates. Table 8 shows language-specific prompt templates for each task.

B Implementation Details

B.1 General Setup

Hyperparameters. Tables 6 and 7 list the hyperparameters in CPT and evaluation, respectively.

Software. We use HF datasets (Lhoest et al., 2021, v3.6.0) for preprocessing, HF transformers (Wolf et al., 2020, v4.52.4), HF peft (Mangrulkar et al., 2022, v0.15.2), FlashAttention-2 (Dao, 2024, v2.7.4) and PyTorch (Ansel et al., 2024, v2.6.0) for training. We use lm-evaluation-harness (Gao et al., 2023, v0.4.8) for IFEval and GSM8K evaluation, alpaca-eval (Li et al., 2023b, v0.6.6) for AE2 evaluation, Ai2 Safety Tool for T3 evaluation,[9] and HF LightEval (Habib et al., 2023, Commit 327071f) for the rest.

Hardware. We mainly use a single AMD MI300X GPU with ROCm 6.4.1 for experiments. Additionally, we use either a single NVIDIA H100 80GB, A100 80GB, or A100 40GB GPU with CUDA 12.9 for evaluation.

Training Cost and Computational Efficiency. A primary advantage of the SSU framework is its efficiency, owing to the one-shot nature of the static importance scoring. We break down the computational overhead into two components:

- **Scoring (Stage 1):** Generating the importance mask is highly efficient. For a 7B model with 500 calibration samples (sequence length 2,048), the scoring process takes approximately 95 seconds on a single AMD MI300X GPU. As this stage primarily involves forward passes to collect activations, it is less compute-intensive than training and can even be offloaded to a CPU if GPU memory is limited.
- **Adaptation (Stage 3):** Unlike dynamic gradient-masking methods (e.g., GMT), SSU utilizes a static mask. This introduces zero additional overhead during the backward pass. In our OLMo 2 7B experiments, the total training time for SSU (34,156s) was essentially equivalent to full fine-tuning (34,979s), with the minor difference attributable to standard hardware variance.

Overall, the pre-computation overhead for SSU represents less than 0.3% of the total training time, making it a nearly "cost-free" intervention relative to standard adaptation.

B.2 Alternative Scoring Method Implementations

SSU-SparseGPT. This method employs a metric from Frantar and Alistarh (2023) that approximates second-order information. The score for any weight θ_{ij} in an input column j is the average squared activation of the corresponding input neuron: $s_{ij} = \mathbb{E}_{x \in \mathcal{D}_{\text{calib}}} x_j^2$.

[9] Following Lambert et al. (2025), we use their forked version: `https://github.com/nouhadziri/safety-eval-fork` (Commit 2920bb8).

SSU-FIM. This method uses the diagonal of the Fisher Information Matrix, which measures output sensitivity to parameter changes (Kirkpatrick et al., 2017). We approximate the Fisher score for a parameter θ_{ij} as the average squared gradient of the negative log-likelihood loss L over $\mathcal{D}_{\text{calib}}$: $s_{ij} = \mathbb{E}_{(x,y)\in\mathcal{D}_{\text{calib}}}(\frac{\partial L}{\partial \theta_{ij}})^2$.

Hyperparameters	Values
Batch size	32
Number of training steps	12,208
Optimizer	adamw_apex_fused
Adam ϵ	1e-8
Adam β_1	0.9
Adam β_2	0.999
Sequence length	512
Learning rate	5e-5
Learning rate scheduler	cosine
Warmup steps	First 5% of steps
Weight decay	0.01
Attention dropout	0.0
Training precision	BF16
HFT, GMT, SSU	
Target freezing ratio	0.5
GMT	
Accumulation interval	4
AdaLoRA	
Target r	8
LoRA α	32
LoRA dropout	0.05
T_{init}	1,000
T_{final}	8,546
δ_t	20
LoRA β_1	0.85
LoRA β_2	0.85
Coefficient of orthogonal regularization	0.5
LoTA	
Mask calibration steps	100
S2FT	
d_{ratio} (Down)	0.015 (equivalent to LoRA $r = 8$)
o_{ratio} (Output)	0.015 (equivalent to LoRA $r = 8$)

Table 6: Hyperparameters for continual pre-training. Values for GMT and AdaLoRA were selected based on our setup, as they were not provided in their respective original papers (Li et al., 2025; Hui et al., 2025).

Parameters	Values
Temperature	0.8
Repetition penalty	1.1
Top k	40
Top p	0.9 (MT, SUM, MTBench) 0.8 (AE2, IFEval, GSM8K)
Sampling	True
Max. generated tokens	128 (MT, SUM) 512 (AE2) 1,024 (MTBench) 1,280 (IFEval) N/A (GSM8K)

Table 7: Parameters for generation tasks. N/A for GSM8K indicates that a model generates text until it detects default stop symbols or reaches its maximum sequence length.

C Supplementary Results

Tables 9, 10, 11, and 12 show performances on English chat and instruction-following benchmarks, English safety alignment benchmark, general English benchmarks, and general target language benchmarks, respectively. Results for IFEval, AE2, MTB, GSM8K, MT, and SUM are averaged across three different runs. The rest are single-run results as they are evaluated in a deterministic-manner.

Task	Language	Template
X-En MT	English	Translate {X: a target language} to English: {sentence} =
	Nepali	नेपालीलाई अङ्ग्रेजीमा अनुवाद गर्नुहोस्: {sentence} =
	Kyrgyz	Кыргызчадан англисчеге которуу: {sentence} =
	Amharic	አማርኛን ወደ እንግሊዝኛ ተርጉም: {sentence} =
	Hausa	Fassara Hausa zuwa Turanci: {sentence} =
	Igbo	Sụgharịa Igbo gaa na Bekee: {sentence} =
En-X MT	English	Translate English to X: {sentence} =
	Nepali	अङ्ग्रेजीलाई नेपालीमा अनुवाद गर्नुहोस्: {sentence} =
	Kyrgyz	Англисчеден кыргызчага которуу: {sentence} =
	Amharic	እንግሊዝኛን ወደ አማርኛ ተርጉም: {sentence} =
	Hausa	Fassara Turanci zuwa Hausa: {sentence} =
	Igbo	Sụgharịa Bekee gaa n'Igbo: {sentence} =
SUM	English	Summarize the following text in English: {text} Summary:
	Nepali	तलको पाठलाई नेपालीमा संक्षेपमा लेख्नुहोस्: {text} सारांश:
	Kyrgyz	Төмөнкү текстти кыргызча кыскача жазыңыз: {text} Кыскача:
	Amharic	የታችኛው ጽሁፍን በአማርኛ አጭር በማድረግ አሳትረኝ።: {text} አጭር መግለጫ:
	Hausa	Taƙaita rubutu mai zuwa cikin Hausa: {text} Taƙaitawa:
	Igbo	Chịkọta edemede a n'Igbo: {text} Nchịkọta:
MRC	English	{context} Question: {question} A. {option A} B. {option B} C. {option C} D. {option D} Answer:
	Nepali	{context} प्रश्न: {question} A. {option A} B. {option B} C. {option C} D. {option D} उत्तर:
	Kyrgyz	{context} Суроо: {question} A. {option A} B. {option B} C. {option C} D. {option D} Жооп:
	Amharic	{context} ጥያቄ: {question} A. {option A} B. {option B} C. {option C} D. {option D} መልስ:
	Hausa	{context} Tambaya: {question} A. {option A} B. {option B} C. {option C} D. {option D} Amsa:
	Igbo	{context} Ajụjụ: {question} A. {option A} B. {option B} C. {option C} D. {option D} Azịza:
MMLU	English	The following are multiple choice questions (with answers) about {subject}. {context} Question: {question} A. {option A} B. {option B} C. {option C} D. {option D} Answer:
	Nepali	तल {subject} सम्बन्धी बहु-विकल्प प्रश्नहरू (उत्तर सहित) दिइएका छन्। {context} प्रश्न: {question} A. {option A} B. {option B} C. {option C} D. {option D} उत्तर:
	Kyrgyz	Бул {subject} боюнча бир нече тандоо суроолору (жооптор менен) төмөндө келтирилген. {context} Суроо: {question} A. {option A} B. {option B} C. {option C} D. {option D} Жооп:
	Amharic	ከታች ስለ {subject} የቀረቡ ባለብዙ ምርጫ ጥያቄዎች (ከመልሶች ጋር) ናቸው። {context} ጥያቄ: {question} A. {option A} B. {option B} C. {option C} D. {option D} መልስ:
	Hausa	Wadannan tambayoyi masu zaɓi da yawa (tare da amsoshi) game da {subject} ne. {context} Tambaya: {question} A. {option A} B. {option B} C. {option C} D. {option D} Amsa:
	Igbo	Nke a bụ ajụjụ ọnụ nhọrọ ọtụtụ (na azịza) gbasara {subject}. {context} Ajụjụ: {question} A. {option A} B. {option B} C. {option C} D. {option D} Azịza:

Table 8: Language-specific prompt templates. We generate the templates for each target language using a machine translation API, following Yong et al. (2023).

	Approach	IFEval					AE2					MTB					GSM8K				
		ne	ky	am	ha	ig	ne	ky	am	ha	ig	ne	ky	am	ha	ig	ne	ky	am	ha	ig
7B	Source	.675	.675	.675	.675	.675	32.6	32.6	32.6	32.6	32.6	3.98	3.98	3.98	3.98	3.98	.796	.796	.796	.796	.796
	FFT	.520	.480	.495	.417	.369	14.3	12.6	12.1	7.8	5.2	3.80	3.50	3.60	3.40	3.12	.623	.619	.593	.602	.604
	AdaLoRA	**.668**	**.679**	**.681**	.646	.669	27.2	25.7	25.7	**24.6**	20.0	3.98	3.96	3.89	**3.92**	3.87	.736	.742	.737	.704	.685
	HFT	.636	.652	.636	.604	.578	22.6	18.3	21.0	15.1	11.1	3.95	3.82	3.85	3.77	3.73	.699	.689	.692	.646	.659
	GMT	.596	.571	.577	.405	.492	17.7	14.2	16.1	7.3	7.3	3.92	3.74	3.79	3.44	3.49	.671	.607	.645	.606	.648
	SSU-Rand	.619	.624	.634	.599	.564	24.0	19.1	19.8	14.8	12.5	3.86	3.81	3.87	3.79	3.75	.701	.678	.693	.660	.680
	SSU-Mag	.595	.617	.591	.548	.497	19.2	16.8	18.3	11.5	8.9	3.87	3.86	3.81	3.79	3.59	.682	.665	.660	.629	.638
	SSU-Wanda	.655	.664	.661	**.688**	**.670**	**28.1**	**28.7**	**28.5**	**24.6**	**25.0**	**4.02**	**4.02**	**3.96**	3.91	**3.92**	**.746**	**.759**	**.749**	**.741**	**.756**
13B	Source	.763	.763	.763	.763	.763	37.2	37.2	37.2	37.2	37.2	4.06	4.06	4.06	4.06	4.06	.853	.853	.853	.853	.853
	FFT	.549	.468	.506	.405	.314	23.6	14.7	18.6	11.9	3.7	3.91	3.66	3.69	3.43	2.93	.768	.730	.732	.733	.737
	AdaLoRA	**.720**	**.733**	**.737**	.728	.675	34.6	**34.1**	**33.2**	30.0	28.7	**4.10**	4.08	**4.09**	4.03	3.94	.812	.814	.812	**.821**	.815
	HFT	.693	.680	.676	.578	.528	31.2	29.1	27.4	23.4	17.9	4.08	4.04	3.99	3.84	3.69	.802	.793	.762	.760	.765
	GMT	.628	.527	.543	.404	.381	28.1	20.1	19.8	16.2	12.3	3.91	3.89	3.54	3.55	3.34	.787	.759	.688	.763	.771
	SSU-Rand	.672	.703	.677	.558	.539	30.2	28.2	26.8	21.9	16.2	3.97	3.97	3.98	3.85	3.66	.787	.795	.777	.766	.780
	SSU-Mag	.651	.648	.636	.489	.434	28.3	24.8	23.5	16.8	9.7	4.00	3.93	3.98	3.76	3.35	.782	.768	.755	.756	.751
	SSU-Wanda	.718	.723	.733	**.739**	**.739**	**34.7**	33.7	32.2	**33.8**	**32.8**	4.04	**4.11**	4.01	**4.10**	**4.01**	**.831**	**.827**	**.814**	.808	**.830**

Table 9: Performance on chat and instruction-following tasks in English. The best and second-best adaptation approaches for each model scale are indicated in **bold** and underlined, respectively.

	Approach	T3 (↑)				
		ne	ky	am	ha	ig
7B	Source	.851	.851	.851	.851	.851
	FFT	.770	.791	.800	.807	.816
	AdaLoRA	**.842**	.829	.836	.806	.805
	HFT	.812	.816	.839	.833	.828
	GMT	.777	.791	.811	.782	.812
	SSU-Rand	.824	.838	.841	.832	.838
	SSU-Mag	.811	.813	.831	.829	.828
	SSU-Wanda	**.842**	**.846**	**.855**	**.856**	**.851**
13B	Source	.821	.821	.821	.821	.821
	FFT	.745	.710	.792	.657	.782
	AdaLoRA	**.816**	**.805**	.815	.759	.799
	HFT	.790	.743	.817	.764	.812
	GMT	.756	.735	.751	.736	.798
	SSU-Rand	.798	.756	.792	.768	.799
	SSU-Mag	.774	.742	.804	.747	.811
	SSU-Wanda	.809	.789	**.819**	**.797**	**.813**

Table 10: Performance on Tülu 3 safety evaluation suite (T3). The best and second-best adaptation approaches for each model scale are indicated in **bold** and underlined, respectively.

	Approach	MT					SUM					MRC					MMLU				
		ne	ky	am	ha	ig	ne	ky	am	ha	ig	ne	ky	am	ha	ig	ne	ky	am	ha	ig
7B	Source	45.4	28.8	19.5	27.9	28.5	22.8	22.8	22.8	22.8	22.8	.880	.880	.880	.880	.880	.618	.618	.618	.618	.618
	FFT	49.5	**44.2**	28.0	48.6	43.6	21.8	20.6	20.1	21.1	20.5	.842	.829	.852	.843	.841	.574	.582	.586	.578	.579
	AdaLoRA	47.6	33.1	14.1	39.8	36.2	22.4	22.9	**22.6**	22.1	22.1	**.874**	**.878**	.871	.860	.847	**.608**	**.614**	**.611**	.585	.593
	HFT	**52.5**	43.7	35.8	48.4	45.4	22.6	22.7	22.0	22.1	22.3	.858	.863	.857	.846	.847	.596	.597	.604	.586	.594
	GMT	50.3	43.7	**37.8**	49.1	**46.7**	22.4	22.2	21.6	20.5	21.5	.850	.818	.856	.829	.853	.579	.578	.599	.565	.591
	SSU-Rand	51.6	44.1	36.4	49.4	45.9	**22.7**	22.8	22.1	22.2	22.4	.858	.864	.872	.856	.856	.600	.599	.605	.584	.597
	SSU-Mag	51.4	43.4	35.8	47.9	45.1	22.5	22.0	21.9	22.1	21.7	.863	.864	.867	.849	.852	.592	.595	.607	.581	.592
	SSU-Wanda	52.3	43.9	36.4	**49.7**	46.3	**22.7**	**23.1**	22.2	**22.9**	**23.3**	.871	.868	**.874**	**.863**	**.870**	.606	.608	.609	**.605**	**.603**
13B	Source	50.7	30.5	22.7	31.0	31.9	24.5	24.5	24.5	24.5	24.5	.897	.897	.897	.897	.897	.665	.665	.665	.665	.665
	FFT	49.7	39.2	39.2	43.5	28.8	21.5	8.6	19.0	14.4	14.8	.890	.891	**.901**	.891	.889	.650	.643	.657	.650	.637
	AdaLoRA	52.1	33.1	19.8	40.6	37.2	24.1	**25.6**	**24.4**	**24.7**	23.4	**.906**	.901	.898	.894	.892	**.662**	**.663**	.662	**.660**	.651
	HFT	55.1	38.6	41.6	50.1	35.1	24.5	20.5	22.7	16.8	18.8	.897	.896	.893	**.899**	.888	.659	.652	**.665**	.657	.655
	GMT	48.7	37.1	23.2	45.2	33.4	23.4	12.9	15.9	14.1	16.4	.892	.893	.900	.896	**.897**	.653	.658	.660	.654	.643
	SSU-Rand	54.4	39.7	36.3	49.7	39.6	**24.9**	23.6	22.9	16.6	20.4	.897	**.903**	.900	.897	.891	.658	.654	.663	.653	.653
	SSU-Mag	53.4	37.4	32.5	45.9	31.5	24.4	20.6	20.7	16.8	18.6	.893	.896	.896	.894	.883	.659	.656	.662	.659	.647
	SSU-Wanda	**55.7**	**45.1**	**43.8**	**51.4**	**45.1**	24.4	25.3	24.0	23.8	**23.8**	.898	.901	.893	.898	**.897**	**.662**	.660	.664	.659	**.659**

Table 11: Performance on source language (English) tasks. Scores that are better than Source are highlighted in green. The best and second-best adaptation approaches for each model scale are indicated in **bold** and underlined, respectively.

	Approach	MT					SUM					MRC					MMLU				
		ne	ky	am	ha	ig	ne	ky	am	ha	ig	ne	ky	am	ha	ig	ne	ky	am	ha	ig
7B	Source	27.0	21.1	5.1	24.4	23.0	22.4	22.9	8.6	23.7	23.3	.382	.379	.276	.332	.301	.301	.301	.276	.321	.323
	FFT	32.5	**33.8**	**12.1**	38.6	36.7	22.1	23.7	9.3	32.2	26.4	.360	.441	.309	**.460**	.396	.293	.312	.288	**.372**	.360
	AdaLoRA	28.1	22.3	4.0	22.9	22.3	21.7	23.1	6.5	31.6	**26.6**	.351	.343	.276	.328	.291	.309	.311	.272	.278	.324
	HFT	32.7	32.4	9.6	37.5	36.9	**22.4**	23.8	8.6	32.1	26.3	.368	.411	.282	.438	.388	.293	.314	.287	.346	**.373**
	GMT	32.3	33.5	11.6	39.0	**38.3**	22.3	23.8	**9.9**	**32.4**	26.2	.346	.419	.312	.451	.398	.279	.308	**.296**	.353	.361
	SSU-Rand	33.2	32.6	9.5	38.4	37.3	**22.4**	23.8	8.8	32.2	26.4	.388	.428	.299	.457	**.401**	.305	.311	.288	.362	.355
	SSU-Mag	33.1	32.2	9.7	37.1	36.6	22.2	23.7	9.2	32.3	26.2	.372	.418	.297	.451	.379	.303	.307	.291	.346	.348
	SSU-Wanda	**34.0**	32.2	9.0	**42.6**	37.1	**22.4**	**24.2**	8.9	32.2	26.3	**.401**	**.458**	**.316**	.439	**.401**	**.313**	**.329**	**.296**	.355	.371
13B	Source	32.4	22.5	6.0	25.3	25.7	22.9	23.2	10.0	25.3	22.4	.501	.393	.318	.348	.310	.345	.322	.293	.333	.351
	FFT	37.5	**36.9**	**16.5**	40.2	37.1	21.8	23.7	10.6	32.7	25.4	.500	**.564**	**.381**	**.579**	.438	.342	.335	.315	**.417**	**.397**
	AdaLoRA	33.7	24.0	5.7	26.3	25.4	22.2	22.9	9.4	31.6	25.4	.448	.391	.293	.371	.322	.340	.307	.277	.324	.307
	HFT	37.6	36.3	14.4	41.6	38.4	21.9	23.4	10.4	32.4	**26.1**	.498	.538	.376	.538	.429	.348	.356	.312	.384	.375
	GMT	37.3	36.6	**16.5**	40.2	36.8	22.0	23.4	9.8	32.7	26.0	.501	.559	.355	.530	.420	.348	.356	**.318**	.404	.338
	SSU-Rand	37.5	36.1	14.5	41.8	37.9	22.3	23.4	10.4	**32.9**	**26.1**	.492	.556	.364	.540	.440	.352	**.361**	.313	.383	.369
	SSU-Mag	37.2	36.1	14.5	39.7	36.5	22.0	23.0	9.7	32.1	26.0	.474	.533	.361	.546	.419	.345	.357	.311	.394	.342
	SSU-Wanda	**37.9**	35.7	13.7	**44.0**	**39.1**	**22.8**	**23.8**	**11.0**	32.3	25.9	**.520**	.549	.377	.542	**.441**	**.354**	.355	.302	.390	.395

Table 12: Performance on target language tasks. Scores that are better than Source are highlighted in green. The best and second-best adaptation approaches for each model scale are indicated in **bold** and underlined, respectively.

D Supplementary Analysis

D.1 Impact of Freezing Ratio on Baselines

We extend this analysis to state-of-the-art selective parameter update baselines (Figure 3). The closest baseline, the static method HFT, follows a trend similar to SSU but fails to surpass the performance of SSU across tasks and freezing ratios. In contrast, the dynamic method GMT exhibits a different trend. While it often achieves strong target language and MT performance at ratios above 60%, it consistently yields low performance on monolingual source tasks regardless of the freezing ratio. We attribute this to the dynamic nature of GMT, which allows updates to any parameter over time, leading to cumulative corruption from unstructured target data optimization (§5). Ultimately, this confirms SSU as the optimal method for simultaneously achieving strong source preservation and high target language gains.

D.2 Calibration Data Size for Parameter Importance Scoring

SSU uses 500 source calibration examples by default to compute parameter importance scores (§4.3). To assess sensitivity to this hyperparameter, we compare the default (500 examples, ∼1M tokens) with a smaller 128-example set (∼0.26M tokens), a size common in model pruning literature (Williams and Aletras, 2024). As shown in Table 13, the results demonstrate minimal changes across tasks; the maximum performance difference observed is only 1.2 points on IFEval. This confirms the robustness of SSU to calibration data size, demonstrating that a small sample set suffices for effective importance scoring.

D.3 Comparison to Additional Baselines

We compare SSU against two other recent selective parameter update methods: LoTA (Panda et al., 2024) and S2FT (Yang et al., 2024). For LoTA, we evaluate both its default 90% sparsity and a 50% sparsity setting that matches the freezing ratio of SSU. For S2FT, we evaluate the default sparsity configuration that sparsely tunes only down-projection layers.

As detailed in Table 14, neither baseline achieves the balanced performance of SSU-Wanda. LoTA at 90% sparsity exhibits inferior source preservation compared to SSU (7.6% vs. 4.0% average drop) and lower target gains (23.9% vs. 30.7%). While reducing LoTA sparsity to 50% improves target gains to 31.7%, it triggers severe catastrophic forgetting, with monolingual source performance dropping by 19.9%. S2FT effectively preserves source capabilities (3.3% drop) but yields negligible target gains (2.3%). These results underscore that only SSU-Wanda simultaneously achieves strong source preservation and high target language gains comparable to FFT.

Sensitivity Analysis. To ensure these findings are not artifacts of specific sparsity choices, we extend our evaluation with a fine-grained ablation study across varying sparsity levels (Table 15).

LoTA: We examine LoTA across sparsity ratios in 12.5% increments. High sparsity configurations (e.g., 90% and 87.5%) preserve source performance reasonably well but consistently underperform SSU-Wanda on both source preservation and target acquisition. Conversely, lowering sparsity allows for more adaptation but disproportionately harms source capabilities. For instance, while LoTA at 50% achieves a 31.7% average target gain, surpassing the 30.7% gain of SSU-Wanda. However, it suffers a drastic 19.9% drop in monolingual source tasks. This degradation worsens at 37.5% sparsity, reaching a 25.4% drop. This confirms that LoTA fails to find an optimal balance between the stability-plasticity trade-off required for effective adaptation.

S2FT: Following the original paper (Yang et al., 2024), we sparsely tune the down projection layers using a parameter count equivalent to LoRA with a rank of 8 (Table 6). We expand the S2FT evaluation by increasing the trainable parameter budget to match LoRA ranks of 16, 32, and 64. We also test the "Down and Output" projection tuning strategy to determine if the poor performance reported for Mistral and Llama3 (attributed to inflexible selection in multi-query attention) applies to OLMo 2. First, increasing the parameter budget improves target performance slightly but erodes source capabilities without ever matching SSU. At the equivalent of rank 64, S2FT suffers a larger source drop (8.2%) than SSU-Wanda (4.0%) while achieving only half the target gains (15.0% vs. 30.7%). Second, we confirm that tuning "Down and Output" projections yields suboptimal results for OLMo 2, causing severe drops of up to 23.1% in source tasks. In summary, regardless of sparsity-level adjustments, only SSU provides robust source preservation while improving target language abilities to levels comparable to FFT.

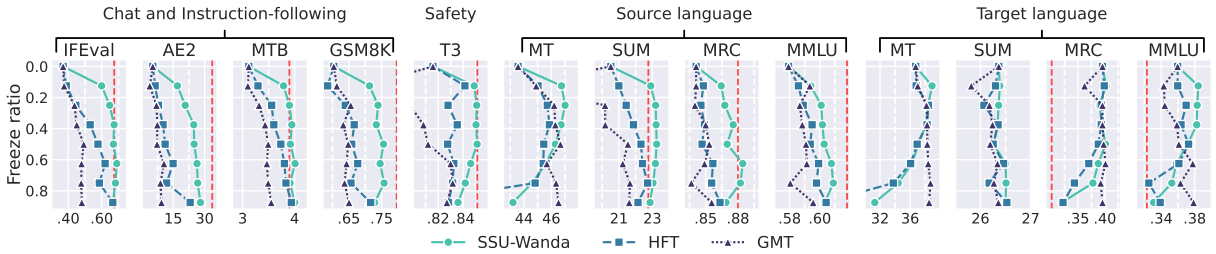

Figure 3: Model performance (SSU-Wanda, HFT, GMT) on Igbo as target language across freezing ratios. The dashed red line indicates Source performance (omitted for MT and SUM due to very low scores). Some data points for HFT and GMT are also omitted due to extremely low performance.

	Chat and Instruction-following				**Safety**	**Source language**				**Target language** (Igbo)			
Approach	**IFEval**	**AE2**	**MTB**	**GSM8K**	**T3**	**MT**	**SUM**	**MRC**	**MMLU**	**MT**	**SUM**	**MRC**	**MMLU**
Source	.675	32.6	3.98	.796	.851	28.5	22.8	.880	.618	23.0	23.3	.301	.323
500 examples (Default)	.670	**25.0**	**3.92**	**.756**	.851	46.3	**23.3**	.870	**.603**	37.1	**26.3**	.401	**.371**
128 examples	**.682**	24.3	3.89	.754	**.852**	**46.4**	23.2	**.873**	.600	**37.2**	**26.3**	**.410**	**.371**

Table 13: Performance of SSU-Wanda with different number of calibration samples. We use Igbo as the target language and tulu-3-sft-olmo-2-mixture as the calibration dataset. **Bold** indicates best adaptation approach.

D.4 Generalization to OLMo 3 Architecture

To evaluate the generalizability of the SSU framework, we measure the performance of the method using the recent Olmo-3-7B-Instruct (Team Olmo et al., 2025), which was released on November 20, 2025. Due to constraints on computational resources, this evaluation focuses on adapting the model to Igbo as the target language. We compare SSU against full fine-tuning (FFT) and all the selective parameter update baselines used in this study.

Results in Table 16 demonstrate that SSU effectively preserves knowledge from the source, yielding an average relative performance degradation of only 1.1% on monolingual source tasks. In comparison, FFT and GMT exhibit substantially higher degradation at 5.9% and 4.5%, respectively. Although S2FT avoids degradation almost entirely (-0.1%), it fails to facilitate adaptation and results in performance in the target language that is 1.4% lower than the original model.

In the target language tasks, SSU achieves average relative gains of 17.3%. While target-driven signals in GMT lead to higher target improvements (20.5%), this approach causes substantially more forgetting than SSU (4.5% versus 1.1%). Furthermore, SSU outperforms static selective parameter update methods such as HFT and LoTA. Both achieve lower target gains (16.5% for HFT and 13.0% for LoTA) and higher source degradation (3.7% for HFT and 3.0% for LoTA).

In summary, SSU achieves the most effective balance by maintaining the general-purpose capability while providing consistent performance gains in the target language. This confirms that SSU remains effective for the recent fully-open instruct model.

D.5 Theoretical Analysis

SSU addresses the stability-plasticity dilemma in neural systems (Grossberg, 1982), balancing plasticity for new knowledge with stability for prior knowledge. By identifying and freezing a source-critical subnetwork, SSU extends the Lottery Ticket Hypothesis (Frankle and Carbin, 2019) to the domain of transfer learning. The use of an importance score to shield crucial parameters enforces a hard constraint, confining updates to a subspace that avoids interfering with source language knowledge. This aligns with recent findings on spurious forgetting (Zheng et al., 2025), which suggest that performance drops often stem from task misalignment caused by nearly orthogonal weight updates.

Furthermore, SSU employs structured, column-wise masking specifically to preserve entire learned features. Unlike unstructured pruning, which can degrade learned representations arbitrarily, pruning entire columns of a weight matrix corresponds to removing specific neurons or feature detectors (Voita et al., 2019). This structural preservation ensures that the core feature space of the source model remains intact, enabling effective adaptation to the target language.

Approach	Chat and Instruction-following				Safety	Source language				Target language (Igbo)			
	IFEval	AE2	MTB	GSM8K	T3	MT	SUM	MRC	MMLU	MT	SUM	MRC	MMLU
Source	.675 +0.0	32.6 +0.0	3.98 +0.0	.796 +0.0	.851 +0.0	28.5 +0.0	22.8 +0.0	.880 +0.0	.618 +0.0	23.0 +0.0	23.3 +0.0	.301 +0.0	.323 +0.0
SSU-Wanda	.670 -0.7	25.0 -23.2	3.92 -1.5	.756 -5.0	**.851** -0.0	**46.3** +62.7	**23.3** +2.3	.870 -1.1	.603 -2.4	37.1 +61.7	26.3 +12.9	.401 +33.2	.371 +14.9
LoTA (90% Sparsity)	.638 -5.4	20.4 -37.4	3.98 +0.0	.706 -11.3	.827 -2.8	45.2 +58.8	22.7 -0.3	.864 -1.8	**.606** -2.0	34.4 +49.9	26.2 +12.5	.366 +21.5	.360 +11.5
LoTA (50% Sparsity)	.449 -33.4	8.3 -74.5	3.45 -13.3	.636 -20.1	.824 -3.2	45.8 +60.9	21.5 -5.6	.844 -4.1	.590 -4.6	**37.8** +64.7	**26.4** +13.4	**.402** +33.5	**.372** +15.2
S2FT (Down)	**.695** +3.0	**27.9** -14.3	**3.99** +0.3	.732 -8.0	.834 -2.0	36.7 +29.0	22.6 -0.7	.857 -2.6	.603 -2.4	21.7 -5.4	26.0 +11.6	.303 +0.6	.331 +2.5

Table 14: Performance of additional adaptation baselines: LoTA and S2FT using Igbo as the target. **Bold** and underlined denote best and second-best adaptation approaches with relative changes (%) in subscripts.

Approach	Chat and Instruction-following				Safety	Source language				Target language (Igbo)			
	IFEval	AE2	MTB	GSM8K	T3	MT	SUM	MRC	MMLU	MT	SUM	MRC	MMLU
Source	.675 +0.0	32.6 +0.0	3.98 +0.0	.796 +0.0	.851 +0.0	28.5 +0.0	22.8 +0.0	.880 +0.0	.618 +0.0	23.0 +0.0	23.3 +0.0	.301 +0.0	.323 +0.0
SSU-Wanda	.670 -0.7	25.0 -23.2	3.92 -1.5	**.756** -5.0	**.851** -0.0	46.3 +62.7	**23.3** +2.3	**.870** -1.1	.603 -2.4	37.1 +61.7	26.3 +12.9	.401 +33.2	.371 +14.9
LoTA (12.5%)	.367 -45.6	5.4 -83.4	3.10 -22.1	.590 -25.9	.811 -4.7	42.1 +47.9	20.4 -10.4	.857 -2.6	.587 -5.0	37.1 +61.7	26.3 +12.9	**.402** +33.5	**.374** +15.8
LoTA (25.0%)	.366 -45.8	5.0 -84.6	3.09 -22.3	.590 -25.9	.812 -4.6	42.2 +48.3	20.4 -10.4	.857 -2.6	.587 -5.0	37.1 +61.7	26.4 +13.4	**.402** +33.5	**.374** +15.8
LoTA (37.5%)	.367 -45.6	4.9 -85.0	3.02 -24.1	.590 -25.9	.811 -4.7	42.5 +49.3	20.4 -10.4	.857 -2.6	.587 -5.0	37.2 +62.1	**26.5** +13.8	**.402** +33.5	**.374** +15.8
LoTA (50.0%)	.449 -33.4	8.3 -74.5	3.45 -13.3	.636 -20.1	.824 -3.2	45.8 +60.9	21.5 -5.6	.844 -4.1	.590 -4.6	37.8 +64.7	26.4 +13.4	**.402** +33.5	.372 +15.2
LoTA (62.5%)	.508 -24.7	8.8 -73.0	3.49 -12.3	.660 -17.1	.832 -2.3	**46.7** +64.1	21.6 -5.1	.853 -3.1	.596 -3.6	**37.9** +65.1	26.4 +13.4	**.402** +33.5	.370 +14.6
LoTA (75.0%)	.573 -15.1	10.2 -68.7	3.76 -5.5	.672 -15.6	.838 -1.6	46.3 +62.7	22.2 -2.5	.853 -3.1	.593 -4.1	37.6 +63.8	26.3 +12.9	.389 +29.2	.369 +14.3
LoTA (87.5%)	.648 -4.0	18.0 -44.7	3.84 -3.5	.681 -14.5	.844 -0.8	45.8 +60.9	22.9 +0.6	.863 -1.9	.603 -2.4	35.1 +52.9	26.2 +12.5	.376 +24.9	.348 +7.8
⋆ LoTA (90%)	.638 -5.4	20.4 -37.4	3.98 +0.0	.706 -11.3	.827 -2.8	45.2 +58.8	22.7 -0.3	.864 -1.8	**.606** -2.0	34.4 +49.9	26.2 +12.5	.366 +21.5	.360 +11.5
⋆ S2FT (Down)	**.695** +3.0	**27.9** -14.3	**3.99** +0.3	.732 -8.0	.834 -2.0	36.7 +29.0	22.6 -0.7	.857 -2.6	.603 -2.4	21.7 -5.4	26.0 +11.6	.303 +0.6	.331 +2.5
S2FT (Down + Output)	.635 -5.9	19.5 -40.1	3.75 -5.7	.306 -61.6	.822 -3.4	30.0 +5.4	21.9 -3.8	.632 -28.2	.393 -36.4	19.7 -14.2	25.3 +8.6	.279 -7.3	.245 -24.1
S2FT (Down; $r = 16$)	.678 +0.5	25.7 -21.1	3.96 -0.5	.735 -7.7	.841 -1.2	38.7 +36.0	22.8 +0.1	.852 -3.2	**.606** -2.0	24.7 +7.6	25.9 +11.2	.314 +4.3	.328 +1.6
S2FT (Down; $r = 32$)	.661 -2.0	21.6 -33.7	3.92 -1.5	.706 -11.3	.837 -1.7	41.7 +46.5	22.7 -0.3	.860 -2.3	.603 -2.4	27.4 +19.4	26.1 +12.1	.316 +4.9	.333 +3.1
S2FT (Down; $r = 64$)	.652 -3.4	19.7 -39.5	3.82 -4.0	.683 -14.2	.846 -0.6	43.2 +51.8	22.9 +0.6	.859 -2.4	.603 -2.4	31.0 +35.1	26.3 +12.9	.317 +5.3	.344 +6.5

Table 15: Ablation study of LoTA and S2FT across varying sparsity levels with Igbo as the target language. **Bold** and underlined denote best and second-best adaptation approaches with relative changes in subscripts. ⋆ indicates the default configuration tested in Table 14.

Approach	Chat and Instruction-following				Safety	Source language				Target language (Igbo)			
	IFEval	AE2	MTB	GSM8K	T3	MT	SUM	MRC	MMLU	MT	SUM	MRC	MMLU
Source	.797 +0.0	29.7 +0.0	4.16 +0.0	.853 +0.0	.786 +0.0	29.4 +0.0	23.6 +0.0	.871 +0.0	.625 +0.0	24.6 +0.0	23.7 +0.0	.329 +0.0	.330 +0.0
FFT	.717 -10.1	28.0 -5.7	4.11 -1.3	.666 -21.9	.778 -1.0	35.0 +19.1	23.7 +0.6	.829 -4.8	.608 -2.7	**34.4** +40.0	26.2 +10.5	**.388** +18.0	.383 +16.0
GMT	.741 -7.1	27.7 -6.7	4.16 -0.1	.719 -15.7	.773 -1.7	35.4 +20.5	**23.8** +1.0	.840 -3.6	.610 -2.4	34.3 +39.6	26.2 +10.5	.379 +15.2	**.385** +16.6
HFT	.780 -2.2	28.9 -2.7	4.11 -1.3	.695 -18.5	.788 +0.3	33.5 +14.0	23.5 -0.2	.847 -2.8	.609 -2.5	32.8 +33.5	26.2 +10.5	.364 +10.7	.368 +11.4
LoTA (90% Sparsity)	.781 -2.0	29.3 -1.3	4.13 -0.8	.723 -15.2	.784 -0.3	33.2 +13.0	23.7 +0.6	.853 -2.1	.608 -2.7	31.8 +29.4	26.0 +9.7	.343 +4.3	.358 +8.4
S2FT (Down)	**.807** +1.2	28.6 -3.7	**4.18** +0.4	**.851** -0.2	**.803** +2.2	28.9 -1.7	23.5 -0.2	**.864** -0.8	**.627** +0.3	24.1 -1.9	22.8 -3.8	.329 +0.0	.330 -0.1
SSU-Wanda	.799 +0.2	**31.0** +4.4	4.17 +0.1	.777 -8.9	.781 -0.6	**37.9** +29.0	23.5 -0.2	.851 -2.3	.618 -1.1	34.0 +38.4	**26.4** +11.4	.357 +8.5	.366 +10.8

Table 16: Performance on Olmo-3-7B-Instruct. We use compare all selective parameter tuning baselines: GMT, HFT, LoTA, and S2FT with SSU-Wanda. We use Igbo as the target language. Relative changes (%) compared to Source are indicated as subscripts. Scores that are better than Source are highlighted in green. The best and second-best adaptation approaches are indicated in **bold** and underlined, respectively.

D.6 Proxy Evaluation on Target-Language Instruction-following

Evaluating instruction-following abilities in under-represented languages is challenging due to data scarcity and unreliable LLM-based judges (Azime et al., 2024). Given these limitations, we establish a tractable proxy for the evaluation.

Specifically, we repurpose machine translation (MT) to serve as a proxy task for instruction-following in a target language. A key aspect of the methodology is instructing models in the target language, not English (Table 8). This design assesses how well a model comprehends and executes instructions within a specific linguistic context, providing a more realistic test of target-language instruction-following. This method thus measures both translation quality and the ability to perform a directed task from instructions in a non-English language.

To quantify performance, we adapt the verifiable evaluation framework of IFEval (Zhou et al., 2023) and its multilingual extension (Zeng et al., 2025). We compute a strict accuracy score,

	Approach	Target-to-English MT					English-to-target MT				
		ne	ky	am	ha	ig	ne	ky	am	ha	ig
7B	Source	.906	.843	.640	.905	.857	.450	.513	.019	.769	.798
	FFT	.785	.876	.525	.870	.670	.014	.399	.006	.022	.398
	AdaLoRA	.902	.809	.321	.871	.835	.083	.225	.000	.061	.035
	HFT	.906	.879	.726	.898	.898	.021	.510	.002	.031	.667
	GMT	**.909**	.873	.706	.919	.909	.015	.463	.003	.101	.859
	SSU-Rand	.904	.877	.744	.929	.907	.084	.545	.009	.108	.760
	SSU-Mag	.897	.873	.735	.831	.857	.015	.471	.005	.006	.581
	SSU-Wanda	.901	**.880**	**.749**	**.956**	**.922**	**.437**	**.557**	**.015**	**.634**	**.906**
13B	Source	.915	.871	.881	.937	.942	.455	.565	.041	.749	.866
	FFT	.821	.749	.681	.729	.318	.012	.171	.004	.038	.240
	AdaLoRA	**.933**	.779	.621	.888	.787	.021	.120	.001	.032	.019
	HFT	.923	.752	.799	.897	.559	.019	.560	.005	.324	.661
	GMT	.826	.713	.411	.758	.521	.011	.252	.011	.087	.255
	SSU-Rand	.910	.773	.687	.888	.685	.011	.499	.003	.387	.521
	SSU-Mag	.888	.721	.615	.793	.447	.019	.382	.002	.099	.254
	SSU-Wanda	.930	**.861**	**.873**	**.945**	**.875**	**.131**	**.599**	**.035**	**.691**	**.884**

Table 17: Instruction-following performance on MT tasks, evaluated using the IFEval framework. IFEval-style strict accuracy is measured against three verifiable criteria: (i) the response is monolingual in the specified language, (ii) it matches the reference sentence count, and (iii) it uses a language-appropriate full stop. The best and second-best adaptation approaches for each model scale are indicated in **bold** and underlined, respectively.

where a response is considered correct only if it satisfies three verifiable criteria: (i) the response is monolingual in the specified language, (ii) the number of sentences matches that of the gold reference, and (iii) the response ends with a language-appropriate full stop.

Implementation. Response generation follows the exact setup described for the MT evaluation in the main paper. We score responses against the three verifiable criteria using the following checks:

- **Response language**: We use GlotLID to compute a normalized confidence score for the specified language in each response (i.e., English for target-to-English MT, target language for English-to-target MT). A response with a score below 0.9 is considered code-mixed and fails this criterion.
- **Sentence count**: For target-to-English MT, we count sentences in both the response and gold reference using the NLTK tokenizer (Bird and Loper, 2004). For English-to-target MT, we count sentences using a regular expression pattern (`[.?!।።]`).
- **Full stop**: A target-to-English response must end with ".". For English-to-target translation, we check for language-specific full stops: "।" or "." for Nepali, "።" for Amharic, and "." for all other languages.

Results and Analysis. Results from the instruction-following evaluation (Table 17) reveal the consistently strong performance of SSU-Wanda. Across languages and both 7B and 13B model scales, SSU-Wanda achieves the highest strict accuracy scores. This superiority is particularly pronounced in the challenging English-to-target direction, suggesting that its proactive, source-driven structured parameter selection strategy is effective for enhancing target-language instruction-following abilities related to formatting.

Nonetheless, a clear performance disparity emerges between the two translation directions. Models consistently achieve higher accuracy on target-to-English tasks compared to English-to-target tasks. This trend demonstrates that while models can comprehend instructions delivered in a non-English language, they more reliably execute those instructions when generating text in English.

The analysis reveals that generating non-English languages under implicit formatting constraints is a primary obstacle for current models. This difficulty likely stems from the adaptation on unlabeled target language data. The unlabeled target language corpus provides weak signals for format-

ting. Furthermore, the evaluation prompts only request translation without explicitly mentioning punctuation (Table 8). Consequently, models learn linguistic patterns but fail to reliably apply specific formatting rules like terminal punctuation in the target language. For instance, we observe that the best-performing SSU-Wanda models achieve only 1.8% (7B) and 3.5% (13B) adherence to the full stop criterion. Therefore, developing methods to improve target-language instruction-following with only unlabeled corpora remains a crucial future research direction. Additionally, we hope this work inspires the development of extensive instruction-following benchmarks for low-resource languages.

E Extended Related Work

SSU addresses the core challenge of continual learning (CL) in machine learning: adapting a model to new tasks while mitigating catastrophic forgetting (Goodfellow et al., 2015; Kirkpatrick et al., 2017). This section situates SSU within the parameter-centric family of CL solutions. These methods protect knowledge at the parameter level, typically without accessing data from the old task for replay. They generally address two fundamental questions: (1) the **Identification Problem**, defining which parameters are critical to a previous task; and (2) the **Protection Problem**, determining the mechanism to enforce protection on those parameters. Parameter-centric approaches largely fall into three categories: **soft, regularization-based** protection; **hard, architectural-based** protection; and **adaptive, hybrid** methods.

Soft Parameter Protection (Regularization-Based). These methods discourage changes to critical parameters by adding a penalty term to the loss function of the new task. Approaches differ primarily in solving the Identification Problem. Elastic Weight Consolidation (EWC) identifies critical parameters via the Fisher Information Matrix diagonal (Kirkpatrick et al., 2017), while Synaptic Intelligence (SI) computes importance online by tracking the cumulative contribution of each parameter to loss reduction (Zenke et al., 2017). Similarly, Memory Aware Synapses (MAS) estimates importance weights based on the sensitivity of the learned function (output function) to parameter changes, eliminating the need for original labeled data (Aljundi et al., 2018). Soft-Masking of Parameter-Level Gradient Flow (SPG) protects knowledge by directly modulating gradient flow with soft masks rather than modifying the loss objective (Konishi et al., 2023). However, such soft constraints often fail under severe distributional shifts (Wang et al., 2023). This limitation becomes particularly acute in our problem setup (i.e., adapting instruct models using unlabeled target language data), where optimization pressure from unlabeled target corpora can overpower regularization penalties.

Hard Parameter Protection (Isolation & Architectural). These methods enforce stability via structural constraints, such as freezing or allocating parameters, to ensure near-zero forgetting. Hard Attention to the Task (HAT) learns a binary mask, forcing gradients to zero for parameters allocated by the mask from any previous task (Serra et al., 2018). PackNet employs an "iterative prune, fix, and retrain" cycle, freezing the surviving "packed" weights and forcing new tasks to utilize only "free" parameters (Mallya and Lazebnik, 2018). Piggyback represents an extreme form, freezing an entire pre-trained backbone and learning new tasks solely by training new binary masks (Mallya et al., 2018).

Adaptive & Hybrid Protection. This emerging class assesses the properties of an incoming task to select a protection strategy dynamically. Context-aware Task-driven (CAT) automatically detects whether a new task resembles previous ones (Ke et al., 2020), applying Hard Protection (binary mask) for dissimilar tasks and Soft Protection (attention) for similar tasks. Parameter Allocation & Regularization (PAR) identifies task relatedness and applies dynamic protection: "easy" tasks are handled via soft regularization, while "difficult" tasks trigger the hard allocation of a new, isolated expert model (Wang et al., 2023). While promising, the application of such dynamic allocation strategies to the specific constraints of LLM language adaptation remains an interesting avenue for future research.

Situating SSU within Continual Learning. SSU adapts these CL principles for the linguistic adaptation of instruct LLMs. We characterize SSU as a source-focused method utilizing static hard parameter protection. Specifically, it resolves the "Identification Problem" via source-data-driven importance scores (e.g., Wanda) and the "Protection Problem" via column-wise structural freezing. While conceptually aligned with hard parameter protection, SSU overcomes specific limitations regarding **problem setting** and **scale**. Foundational

CL methods largely focus on task-incremental learning, where the model learns a sequence of discrete, labeled tasks (e.g., Task 1: MNIST, Task 2: CIFAR). Consequently, methods like HAT rely on task identifiers (Task IDs) at inference time to select the correct mask. This requirement is incompatible with general-purpose instruct LLMs, where the input language (or task) is unknown and the model must operate as a unified entity without external task signals. Regarding scale, foundational methods typically target architectures with fewer than 1B parameters (e.g., PackNet uses VGG-16 ($\sim$138M) (Simonyan and Zisserman, 2015)). Methods like the iterative pruning and retraining cycles of PackNet often become computationally prohibitive when applied to billion-parameter LLMs. In contrast, SSU utilizes a one-shot, static calculation of importance before training, making it computationally viable for modern transformer-based architectures.

F License

This study uses publicly available models and datasets with different licenses, as detailed below. Note that all permit their use for academic research.

Model Licenses. The OLMo 2 family of models are distributed under Apache License 2.0.

- 7B: https://huggingface.co/allenai/OLMo-2-1124-7B-Instruct
- 13B: https://huggingface.co/allenai/OLMo-2-1124-13B-Instruct

Olmo-3-7B-Instruct is also distributed under Apache License 2.0: https://huggingface.co/allenai/Olmo-3-7B-Instruct.

Data Licenses. tulu-3-sft-olmo-2-mixture is licensed under ODC-BY-1.0. MADLAD-400 is licensed under CC-BY 4.0. XL-Sum is licensed under CC BY-NC-SA 4.0. Belebele and FLORES-200 are licensed under CC BY-SA 4.0. MMLU, GSM8K, and HumanEval are distributed under the MIT License. Ai2 Safety Tool, AlpacaEval, IFEval, and MT-Bench are distributed under Apache License 2.0.

G Use of Generative AI Tools

The authors acknowledge the use of LLMs during the preparation of this work. Gemini 2.5 and 3.0 Pro were utilized to find related work and to improve the grammar and clarity of the draft. Additionally, GPT-5 served as a coding assistant for implementation and debugging.